UNISYS

Firmware Download

Format Specification

3486 8968 Revision B

April 12, 1996

Preliminary
Specification
Layout

for Upgrade and new
program for DFAST

APRIL 1996

E 1

DOCUMENT CONTROL INFORMATION

CEDB controls the latest revision of this document. Consult Engineering Records of the DCL for assistance.

Engineering signatures appear in the EIR releasing this document.

CONTENTS

TABLES

1. SCOPE

This document outlines in detail the format specification of download firmware file format used by Unisys to download firmware.

Revision B enables the ability to determine precisely what type of download command sequence a device may require. The modifications for Revision B do not require a new utility for existing download files unless the utility checks for non-blank data after the last Drive ID entry.

This specification is not meant to define vendor unique tape loading or other non-Write Buffer command based loading methods

2. PURPOSE

The purpose of this document is to standardize the format of the firmware file which will simplify the task of reading the firmware file by the system application programs.

In addition, by standardizing the format of the download firmware file, other future peripheral vendors with download features could be supported by Unisys with minimum modifications required to the system application program. This document supersedes internal document R210.

3. PROCEDURE

SCSI Firmware Download File Format

The firmware data will be provided on a DOS 3.5 " diskette (1.44MB) in binary format. Physically, the media will contain 8K (1K=1024) byte contiguous records. The file shall have two logical record types: a header record which will contain all the information needed to interpret the content and perform some checks on the drives, and a body record which will contain all the firmware code. Physically, the first 8K record on the media will contain the header record. All other 8K records on the media will contain the firmware information. If the firmware size is not a multiple of 8K, then the last 8K record will contain extra storage (00 hex).

4. FORMAT

4.1 File Header Record

The 8,192 byte firmware file header record format will be as follows:

Field Name	Size (in bytes)	Offset
Customer ID	8	0
Size of header record	8	8
Size of firmware data	8	16
New firmware level	4	24
Vendor ID	8	28
Number of drive ID	4	36
Number of replaceable firmware levels	4	40
Firmware level #1 record (most recent)	8	44
Firmware level #2 record	8	52
... (until the last level)	8	...
Drive ID #1	16	... +8
Drive ID #2	16	... + 16
... (until the last Drive ID #n)	16	... +... + 32
MODE	4	... +... + 48
Mode-Start	1	... +... + 52
Mode-Last	1	... +... + 53
Mode Offset Minimum Increment	8	... +... + 54
Mode Offset Maximum Increment	8	... +... + 62

Example Header

```
0:   UNISYS 00008192
16:  01048828 2 bFUJI
32:  TSU 00010001*
48:   2 AM2488D M248
64:  8  MODE67000005
80:  1201048848
```

*Note: The * in the above Header example is supposed to be the value 00 in hex, but there is no way in the ASCII command set to denote that character for the purposes of the example. Also, the above example is for the non-default case of a non-Mode 5 download sequence.*

Unused Bytes in the 8K Header

Typically, there will be a significant amount of unused data at the end of the Header Record. This unused data should be set to 00 hex. Further it is recommended that at least 16 bytes after the last valid entry of header data be set to 00 hex to allow for a break point in the flow of header information.

4.2 Field Definitions- Default

The default case for the download sequence uses a single Write Buffer command with Mode 5.

All letter representation is in upper case.

All data is in ASCII format unless otherwise indicated.

Customer ID

This field is always set to "UNISYS" and is left justified. Unused portions are ASCII blanks 20h.

EXAMPLE: "UNISYS "

Size of Header Record

This field is always set to 8K bytes and is right justified with leading zeros.

EXAMPLE: "00008192"

Size of Firmware Data

This field contains the number of decimal bytes (represented in ASCII) of firmware data and is right justified with leading zeros.

EXAMPLE: "00305152"

This value would convert to 4A800 hex bytes for use in the CDB.

New Firmware Level

Level of firmware as described in bytes 32 through 35 of SCSI INQUIRY data.

EXAMPLE: "0030"

Vendor ID

The name of the drive manufacturer, typically UNISYS for disk products, which corresponds to data returned from bytes 8 through 15 of drive SCSI INQUIRY command. This is specified in the procurement specification for that device and is left justified. Unused portions are ASCII blanks 20h.

EXAMPLE: "UNISYS "

Number of Drive ID

Number of different drive model numbers that use the same firmware and is right justified with leading zeros.

EXAMPLE: "0002"

Number of Replaceable Firmware Levels

Number of all prior firmware levels that were approved for production. This field is right justified with leading zeros.

EXAMPLE: "0002"

NOTE TO IMPLEMENTORS OF DOWNLOADING UTILITY:

If you are downloading a current version on a drive already containing that version of firmware, the use of the "New firmware Level" field data is recommended for verifying the correct version of firmware that is downloaded to the drive.

Firmware Level #X Record Format

8 bytes: numbered 7..0 with left most being byte 7.

byte 7: a format flag meaning,

if = "Hex 00"

"No format needed after the download of this level of firmware", or not a disk device

else

"Format needed after the download" (disk devices only)

byte 6..4:

Reserved (ASCII blanks 20h)

byte 3..0:

firmware level, as it appears in bytes 32 through 35 of the SCSI Inquiry data (Right Justified) for firmware level #X.

Note: Because the firmware level in this field is used to determine whether or not new firmware needs to be loaded, each new release of drive firmware must be uniquely identifiable in bytes 32 through 35 of the SCSI Inquiry data.

Drive ID # Record

Represents the drive model number or "PRODUCT IDENTIFICATION FIELD" as returned from bytes 16 through 31 of SCSI INQUIRY command and is left justified (ASCII blank 20h for unused portions). Unisys will advise the vendor on required drive ID's (Drive model numbers).

EXAMPLES: "ST41600N" or "ST41600ND"

4.3 Field Definitions - Non-Mode 5 Based

The following field definitions are defined for drives that cannot download using a single ANSI standard Write Buffer command with Mode 5. The use of the fields in this section is optional. These fields should not be used if the product can operate with the default download sequence that uses a single Write Buffer command and Mode 5.

NOTE TO IMPLEMENTORS OF DOWNLOADING UTILITY:

If using the non-Mode 5 method, the download utility will have to get the Mode information prior to building the CDB.

MODE Record

This is an ASCII 4 byte identifier for the start of the mode definition section of the header record. If present, it indicates the drive does not support the default download method. The remaining MODE supporting records must be used to determine the appropriate Write Buffer command parameters and sequence.

DATA: "MODE"

Mode-Start

This ASCII 1 byte value indicates what mode of download should be used for all but the last Write Buffer command, in a multiple Write Buffer series to download microcode.

EXAMPLE: "6"

If the drive cannot support multiple Write Buffer commands firmware download and also cannot support the default of a Mode 5 download, then this field shall contain the same value as the Mode-Last field.

Mode-Last

This ASCII 1 byte value indicates what mode of download should be used for the last (or only) Write Buffer command.

EXAMPLE: "7"

Mode Offset Minimum Increment

This 8 byte field identifies the minimum incremental number of decimal bytes (represented in ASCII) of firmware data (offset) the drive will accept for multiple Write Buffer commands. It's format is right justified with leading zeroes.

This value is also used to determine the factor by which an offset greater than the minimum is determined. An offset must be an even multiple of the minimum increment. For example, if the minimum increment is 4096, only offsets in multiples of 4096 (e.g. 8192, 12288, or 65536, etc.) may be used.

EXAMPLE: "00004096"

If the drive does not support the use of offsets in the Write Buffer command, this field shall be set to the Size of Firmware Data field value for consistency.

Mode Offset Maximum Increment

This field identifies the maximum incremental number of decimal bytes (represented in ASCII) of firmware data (offset) the drive will accept for multiple Write Buffer commands. It's format is right justified with leading zeroes.

If the drive does not support the use of offsets in the Write Buffer command, this field shall be set to the Size of Firmware Data field value for consistency.

If the drive can accept a single Write Buffer command for download, this field shall contain the Size of Firmware Data field value for consistency.

EXAMPLE: "00305152"

4.4 Firmware Binary Data

This record immediately follows the 8K file header. It contains the actual firmware code. The size of the field is set the value in the Size of Firmware Data field.

5. APPENDIX

This section does not apply to the Supplier that is trying to build a header and binary code file. It is informational only and intended to explain how the information in the header is used. .

5.1 Method of Downloading Firmware

The SCSI Write Buffer command shall be used to download and save the firmware on the drive. There are two methods for downloading the firmware. The preferred capability is for the drive to receive the entire firmware file body as a single Write Buffer command. The second method utilizes offsets and modes as defined in the SCSI-3 Write Buffer command to enable the firmware file to be sent in multiple commands.

5.2 Default Download

The following download parameters are defined for backwards compatibility with previous Unisys download utilities, and firmware files. This CDB must be used when the MODE field is not present.

The Write Buffer CDB is set as follows, and the firmware is transferred in a single command:

0x3B	
0x05	05 = Microcode Download
0x00	Buffer ID
0x00 0x00 0x00	Reserved
0xXX 0xXX 0xXX	Number of bytes sent to drive (3 byte field=Size of Firmware in Hex)
0x00	

5.3 Mode and Offset Controlled Download

A download using the Mode and Offset controls requires the utility to generate a CDB based on the MODE and increment values. Consider the following MODE header information:

Field	Value
MODE	MODE
Mode-Start	6
Mode-Last	7
Minimum Incr.	00004096
Maximum Incr.	00305152

Note: For the purposes of this example, the maximum increment was set to the Size of Firmware Data value since the drive is capable of a single Write Buffer command download.

The system for which the download will occur can only send a maximum of 65536 bytes in a SCSI command, despite the drive's ability to receive the firmware in a single command. Since this is the case, the utility will have to generate four CDBs with Mode 6 (the Start value), and another CDB with Mode 7 (Last).

The first 4 CDBs contain the following data:

Table 5-1. Download with Offset, CDB #1

0x3B	
0x06	6 = Download with Offset
0x00	Buffer ID
0x00 0x00 0x00	Buffer Offset
0x01 0x00 0x00	Number of bytes sent to drive (3 byte field)
0x00	

Table 5-2. Download with Offset, CDB #2

0x3B	
0x06	6 = Download with Offset
0x00	Buffer ID
0x01 0x00 0x00	Buffer Offset
0x01 0x00 0x00	Number of bytes sent to drive (3 byte field)
0x00	

Table 5-3. Download with Offset, CDB #3

0x3B	
0x06	6 = Download with Offset
0x00	Buffer ID
0x02 0x00 0x00	Buffer Offset
0x01 0x00 0x00	Number of bytes sent to drive (3 byte field)
0x00	

Table 5-4. Download with Offset, CDB #4

0x3B	
0x06	6 = Download with Offset
0x00	Buffer ID
0x03 0x00 0x00	Buffer Offset
0x01 0x00 0x00	Number of bytes sent to drive (3 byte field)
0x00	

The last CDB contains Mode 7, Download with Offset and Save:

Table 5-5. Download with Offset and Save, Last CDB

0x3B	
0x07	7 = Download with Offset and Save
0x00	Buffer ID
0x04 0x00 0x00	Buffer Offset
0x00 0xA8 0x00	Number of bytes sent to drive (3 byte field)
0x00	

5.3 Determining the Method of Download

This section provides a sample algorithm that the utility implementor can use to determine what type of download should be performed based on the information provided in the Header. It is assumed that before this algorithm is processed that the Header data has been validated with the device, i.e. Unisys name in the header, firmware levels compared, and product model compared.

DEFINE VARIABLE "SYSCAP" = The maximum length of a SCSI I/O that the system can send for the Write Buffer Command.

IF THE WORD 'MODE' IS NOT PRESENT IN THE HEADER DATA
THEN ; DEFAULT DOWNLOAD METHOD MUST BE USED
    IF 'Size of firmware data' > SYSCAP
    THEN
        ERROR "DOWNLOAD IS NOT POSSIBLE"
    ELSE
        PERFORM DEFAULT DOWNLOAD (MODE 5, SINGLE COMMAND)

ELSE ; DEFAULT CAN'T BE USED BECAUSE MODE IS PRESENT
    IF SYSCAP ≥ 'Mode Offset Maximum Increment'
    THEN
        IF 'Mode Offset Maximum Increment' = 'Size of firmware data'
        THEN
            "DO 1 I/O WITH 'Mode Last' IN THE CDB, AND LENGTH =
            'Size of firmware data'
        ELSE
            "DO 'n' I/Os WITH 'Mode Start' IN THE CDB, AND LENGTH =
            'Mode Offset Maximum Increment' UNTIL THE LAST I/O WHICH IS
            EXECUTED WITH 'Mode Last' IN THE CDB. DON'T FORGET THAT
            THE OFFSETS IN THE CDB HAVE TO INCREMENT WITH EACH I/O."

    ELSE ; THIS IS THE CASE WHERE DRIVER FIRMWARE IS BIGGER THAN HOST CAP
        IF 'Mode Offset Minimum Increment' > SYSCAP
        THEN
            ERROR "DOWNLOAD IS NOT POSSIBLE. SYSTEM NOT CAPABLE"
        ELSE
            "CALCULATE THE LARGEST POSSIBLE BLOCK SIZE THAT IS A
            MULTIPLE OF THE 'Mode Offset Minimum Increment', BUT IS LESS
            THAN THE SYSCAP SIZE. THEN--
            "DO 'n' I/Os WITH 'Mode Start' IN THE CDB, AND LENGTH =
            'Newly calculated value < SYSCAP' UNTIL THE LAST I/O WHICH IS
            EXECUTED WITH 'Mode Last' IN THE CDB. DON'T FORGET THAT
            THE OFFSETS IN THE CDB HAVE TO INCREMENT WITH EACH I/O."

Alanis, Belisario MV

From:	Leisz, Frank J	TR
To:	Alanis, Belisario	MV; Shelton, Phil MV
Cc:	Hale, Wesley	MV; Bauman, David A; Stell, Jeffery A
Subject:	Test Unit Ready	
Date:	Friday, September 13, 1996 2:09PM	

Hi;

In order to accommodate the SSD request for the addition of the Test Unit Ready command, S/W will continue to use the current interface between the DFAST application and the MCP procedure USERMAINTERQUEST.

To invoke the Test Unit Ready command, the User will pass 11 as the function code. The remaining parameter definitiions are the same.

For this function code, the MCP will issued the Test Unit Ready Command (00h) and return any detected error condition in the parameter DEVICERD.

Any problems or comments on this interface?

TIA

SEPT 1996
Information to
Inventor B. Alanis for
integration with MCP
during DFAST upgrade
development

E2

Alanis, Belisario MV

From: Leisz, Frank J
To: Alanis, Belisario MV
Cc: Houseman, Gloria R; Stell, Jeffery A
Subject: Bootleg 423A MCP (Download Microcode with multi-IO's)
Date: Tuesday, November 19, 1996 1:56PM

Greetings:

I created a new bootleg 423A MCP, which contains the new download function codes and the Ralph's OST patches.

The name of the bootleg MCP is:

(TRANSFER)SYSTEM/423A/MCP/ASDX/UMR ON TRANSFER located on TRPROGD

Hopefully this helps with available machine time, if there are any problems, please call me at sn-385-4487. TIA

Frank

NOV 1996

Information to Inventor B. Alanis for integration with Unisys Master Control Program (MCP)

E3

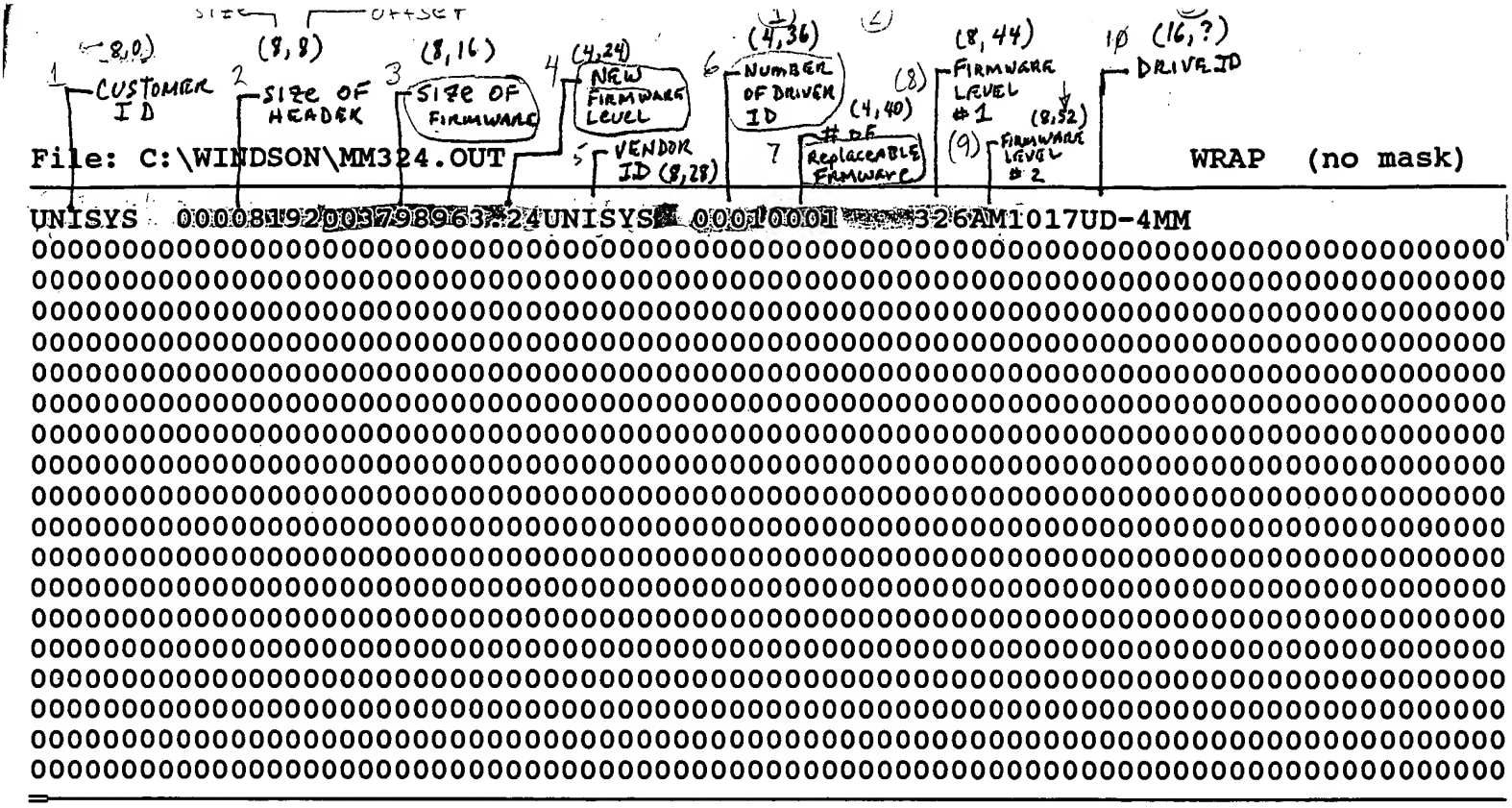
DFAST= ON ENGDATA

File Name	Filekind	Records	Sectors	CreationTime
DFAST	DCALGOLCODE	270	270	08/09/1996
DFAST/NX	DCALGOLCODE	270	270	11/11/1996
DFAST/GAM/DLT/MLTBLK/TUR/NFWCOMPR	DCALGOLCODE	290	297	03/07/1997
DFAST/SPO	DCALGOLCODE	280	288	07/10/1996
DFAST/CHUNK	DCALGOLCODE	270	270	09/23/1996
DFAST/GAMMA/DELTA	DCALGOLCODE	270	270	02/18/1997
DFAST/GAMMA/DELTA/MULTBLK/TUR	DCALGOLCODE	290	297	02/18/1997
DFAST/NXTUR	DCALGOLCODE	290	297	01/28/1997
DFAST/CHUNKY	DCALGOLCODE	286	288	11/27/1996
DFAST/CHUNKY/UNTIL0	DCALGOLCODE	286	288	12/02/1996
DFAST/3DIGIT	DCALGOLCODE	269	270	06/27/1996
DFAST/3DIGIT/SYSTPFORDEV	DCALGOLCODE	270	270	06/27/1996
DFAST/CHUNKY2	DCALGOLCODE	290	297	12/12/1996
DFAST/CHUNKY2/UNISYS	DCALGOLCODE	290	297	01/13/1997
DFAST/NXTURFX	DCALGOLCODE	290	297	01/29/1997

Experimental And Test Runs by Inventor B. Alanis during developments For upgrade to DFAST.

1996 - 1997

(E4)



MVAISA 192.59.221.56

Format Spec. # 34868968 Rev B 4/12/96

Symbol/DFAST/1005000

Header Work in April 1996 by Inventor working on upgrade to DFAST

(E5)

```
(BEL)DFAST ON ENGDATA. (Version 01.002) compiled 08/09/96 @ 15:50, run on 02/20/97 @ 13:52

DO YOU WANT TO VIEW THE USER DOCUMENTATION FOR DFAST? 'Y' OR 'N'
Y
DFAST is a firmware download utility for SCSI disk drives
and/or SBC Controllers.  DFAST requires:

   1.)  Firmware input file on disk or tape; e.g.,

      (UCODE)FWFILE/123 ON ANYPACK - if on disk
                 or
      (UCODE)FWFILE/123            - if on tape

   2.)  A target (SCSI disk drive or SBC Controller)
        that has been reserved (URed).

   3.)  '<Mix #> AX' when responding to a prompt.

The process follows the following format:

A.) The user is asked to select one of four options:
  1.) 'Ctlload'    - select a SBC unit type.
  2.) 'Diskload'   - select a SCSI disk drive unit type.
  3.) 'Verifyfile' - check and display firmware file
                     header information without
                     performing a download to the target.
  4.) 'Quit'       - exit DFAST.
B.) The user is prompted for a firmware file name or
    'Quit' to exit DFAST.
C.) If the file is validated as a firmware file, the new
    firmware level is displayed; otherwise the program
    returns to step 'B'.
D.) If 'Ctlload' or 'Diskload' was entered in step 'A',
    then for every SCSI disk or SBC unit number entered,
    steps 'a' through 'e' below are performed.
a.) Target attributes are obtained and displayed.
b.) If download is not allowed (due to mismatch of file/
    target Firmware level, Product ID or Vendor ID), the
    process stops for that unit, and the user is asked
    to enter another target number of the same unit type
    or 'End' to return to step 'A'.
    otherwise:
c.) If the SCSI drive (not SBC) requires formatting(IVR)
    after the download, the user is consulted to:
  i.)  continue with the firmware download  -or-
  ii.) stop the process for that unit (to back up data
       on that drive) and select another target of the
       same unit type, or 'End' to return to step 'A'.
d.) After the download is complete and the target has
    sequenced, the INQUIRY command is reissued, and the
    new firmware level of the target is displayed.
e.) The user is prompted to enter another target number
    of the same unit type or 'End' to return to step 'A'.

>>>>>>>>>>>>>>>>>>> ATTENTION <<<<<<<<<<<<<<<<<<<
 READ the appropriate FCN for this firmware file
 before downloading the firmware
>>>>>>>>>>>>>>>>>>> ATTENTION <<<<<<<<<<<<<<<<<<<

 Do you want to continue?  Enter 'YES' or 'NO'
```

DFAST upgrade work for Debug & Test by Inventor Alan S in Feb 1997

E6

U N I S Y S

ENGINEERING INFORMATION RELEASE
SUMMARY SHEET

EIR Number Prt Cmp Rv
MV - 200523 001 A

Rel Date: 19990707
Rev Date:

Product Name: CONTRLR SUPPT
Project No : COMMON
Type : CHANGE
Category : SOFTWARE
Eng Affected: MV
Mfg Affected:
REA(s) :

RCC: 4
Priority: A

Page 1 of 23
Summary Shts: 3
Rep/Add Shts: 1
Attachments :

WORKORDER(s): MV44003

Keywrd:
Reason: UPDATE DFAST LVL 1.004 TO LVL 1.005
Explanation - Full:
RELEASES DFAST MEDIA LEVEL 1.005 TO THE PTDTOQUAL KIT AND
RELEASES MEDIA TO POSM. DFAST LEVEL 1.005 ADDS SUPPORT TO THE
NX5820 LVD AND SERVER DOWNLOAD.
34925156-003 TO -004
34925149-003 TO -004

EIR Revision: A Est Date: 19990629 Rel Date: 19990707
Preparer: C. LAKE EMI Coord:
Checker : EMI Info :
Init : Mfg Rev :
Safety : Other :
Authoriz: Other :
Authorizer's Verification Statement:

Lo. Us.	Base/Part	List Fr	List To	Part Fr	Part To	Pic Fr	Pic To	Description			C C	BC St	No. SF.	Pg.	RM
	3492 4639	F	G			*	*	DFASTMT MED ASSY M42				PL	2	4	
	3492 4639-003			F	G			DFASTMT MED ASSY M42			C	I			
	3492 4639-004				G			DFASTMT MED ASSY M42		1.0	N	A			
	3492 5149	F	G			*	*	DFASTQIC MED ASSY M17				PL	2	6	
	3492 5149-003			F	G			DFASTQIC MED ASSY M17			C	I			
	3492 5149-004				G			DFASTQIC MED ASSY M17		1.0	N	A			
	3492 5156	F	G			*	*	DFASTHIC MED ASSY M48				PL	2	8	
	3492 5156-003			F	G			DFASTHIC MED ASSY M48	LV	1.0	C	I			
	3492 5156-004				G			DFASTHIC MED ASSY M48	LV	1.0	N	A			
	3493 0222	E	G			*	*	DFASTHIC MSTR MED M48				TL	2	10	
	3493 0222-003			E	G			DFASTHIC MSTR MED M48			C	I			
	3493 0222-004				G			DFASTHIC MSTR MED M48			N	A			
	3493 0230	E	G			*	*	DFASTHIC LAB TXT M48				TL	2	12	
	3493 0230-003			E	G			DFASTHIC LAB TXT M48			C	I			
	3493 0230-004				G			DFASTHIC LAB TXT M48			N	A			
	3493 1014	E	F			*	*	DFASTQIC MSTR MED M17				TL	2	14	
	3493 1014-003			E	F			DFASTQIC MSTR MED M17			C	I			
	3493 1014-004				F			DFASTQIC MSTR MED M17			N	A			
	3493 1022	E	F			*	*	DFASTQIC LAB TXT M17				TL	2	16	
	3493 1022-003			E	F			DFASTQIC LAB TXT M17			C	I			
	3493 1022-004				F			DFASTQIC LAB TXT M17			N	A			

EXHIBIT

E-8

U N I S Y S

ENGINEERING INFORMATION RELEASE
SUMMARY SHEET

EIR Number Prt Cmp Rv
MV - 200523 001 A

Rel Date: 19990707
Rev Date:

Product Name: CONTRLR SUPPT
RCC: 4 Priority: A

Lo. Us.	Base/Part	List Fr	List To.	Part Fr	Part To.	Pic Fr	Pic To.	Description	C C	BC St	No. SF.	Pg.	RM
	3493 1030	E	F			*	*	DFASTMT MSTR MED M42		TL	2	18	
	3493 1030-003			E	F			DFASTMT MSTR MED M42	C	I			
	3493 1030-004				F			DFASTMT MSTR MED M42	N	A			
	3493 1048	E	F			*	*	DFASTMT LAB TXT M42		TL	2	20	
	3493 1048-003			E	F			DFASTMT LAB TXT M42	C	I			
	3493 1048-004				F			DFASTMT LAB TXT M42	N	A			

EIR CONTINUATION REV: A
RELEASES DFAST MEDIA LEVEL 1.005 TO THE PTDTOQUAL KIT AND
RELEASES MEDIA TO POSM. DFAST LEVEL 1.005 ADDS SUPPORT TO THE
NX5820 LVD AND SERVER DOWNLOAD.
34925156-003 TO -004
34925149-003 TO -004
34924639-003 TO -004
SCRAP OLD PART NUMBERS.

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3492 4639	F G	* *	DFASTMT MED ASSY M42	PL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

Part	Part Fr/To	Description	CC	St
3492 4639-003	F G	DFASTMT MED ASSY M42	C	I

	From	To
CHG Desc:	DFASTMT LV:1.004 ..	DFASTMT MED ASSY M42
CHG Status:	A	I
ADD Newpn:		3492 4639-004

DIFFERENCES BETWEEN 3492 4639-003 REPLACED-INACTIVE BY 3492 4639-004

Find	Ch	Alt		From	To
1					
CHG			Part:	3937 6975-000 MAGNETIC TAPE, 1200 FT. BLANK MEDIA ASSY	3937 7007-000 MAG TAPE 3600 FOOT
DEL	R001		Ref:	3493 1030-003 DFASTMT MSTR MED M42 DFASTMT MSTR MED M42	
	R002				
CHG			Ref:	3493 1048-003 DFASTMT LAB TXT M42 DFASTMT LAB TXT M42	3493 1030-004 DFASTMT MSTR MED M42 DFASTMT MSTR MED M42

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3492 4639	F G	* *	DFASTMT MED ASSY M42	PL	A	4	2

DIFFERENCE REPORT Continued

Find	Ch	Alt	From	To
ADD R003				
	Ref:			3493 1048-004 DFASTMT LAB TXT M42 DFASTMT LAB TXT M42

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description		CC	St
3492 4639-004	G	DFASTMT MED ASSY M42	1.005	N	A

	From	To	
ADD Desc:		DFASTMT MED ASSY M42	1.005
ADD Status:		A	

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST _____ AT VENDOR _____ IN FIELD

COMMENTS:

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3492 5149	F G	* *	DFASTQIC MED ASSY M17	PL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC	St
3492 5149-003 F G	DFASTQIC MED ASSY M17	C	I

	From	To
CHG Desc:	DFASTQIC LV:1.004 ..	DFASTQIC MED ASSY M17
CHG Status:	A	I
ADD Newpn:		3492 5149-004

DIFFERENCES BETWEEN 3492 5149-003 REPLACED-INACTIVE BY 3492 5149-004

Find	Ch	Alt	From	To
DEL R001				
	Ref:		3493 1014-003 DFASTQIC MSTR MED M17 DFASTQIC MSTR MED M17	
R002				
CHG	Ref:		3493 1022-003 DFASTQIC LAB TXT M17 DFASTQIC LAB TXT M17	3493 1014-004 DFASTQIC MSTR MED M17 DFASTQIC MSTR MED M17
ADD R003				
	Ref:			3493 1022-004 DFASTQIC LAB TXT M17 DFASTQIC LAB TXT M17

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3492 5149	F G	* *	DFASTQIC MED ASSY M17	PL	A	4	2

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description		CC	St
3492 5149-004	G	DFASTQIC MED ASSY M17	1.005	N	A

	From	To
ADD Desc:		DFASTQIC MED ASSY M17 1.005
ADD Status:		A

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST _____ AT VENDOR _____ IN FIELD

COMMENTS:

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3492 5156	F G	* *	DFASTHIC MED ASSY M48	PL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description	CC	St
3492 5156-003	F G	DFASTHIC MED ASSY M48 LV 1.00	C	I

	From	To
CHG Desc:	DFASTHIC LV:1.004 ..	DFASTHIC MED ASSY M48 LV 1.00
CHG Status:	A	I
ADD Newpn:		3492 5156-004

DIFFERENCES BETWEEN 3492 5156-003 REPLACED-INACTIVE BY 3492 5156-004

Find	Ch	Alt	From	To
DEL R001				
	Ref:		3493 0222-003 DFASTHIC MSTR MED M48 DFASTHIC MSTR MED M48	
R002				
CHG	Ref:		3493 0230-003 DFASTHIC LAB TXT M48 DFASTHIC LAB TXT M48	3493 0222-004 DFASTHIC MSTR MED M48 DFASTHIC MSTR MED M48
ADD R003				
	Ref:			3493 0230-004 DFASTHIC LAB TXT M48 DFASTHIC LAB TXT M48

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3492 5156	F G	* *	DFASTHIC MED ASSY M48	PL	A	4	2

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description	CC	St
3492 5156-004	G	DFASTHIC MED ASSY M48 LV 1.005	N	A

	From	To
ADD Desc:		DFASTHIC MED ASSY M48 LV 1.005
ADD Status:		A

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST ____ AT VENDOR ____ IN FIELD

COMMENTS:

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 0222	E G	* *	DFASTHIC MSTR MED M48	TL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description	CC	St
3493 0222-003	E G	DFASTHIC MSTR MED M48	C	I

	From	To
ADD Desc:		DFASTHIC MSTR MED M48
CHG Status:	A	I
ADD Newpn:		3493 0222-004

	Part Fr/To	Description	CC	St
3493 0222-004	G	DFASTHIC MSTR MED M48	N	A

	From	To
ADD Desc:		DFASTHIC MSTR MED M48
ADD Status:		A

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 0222	E G	* *	DFASTHIC MSTR MED M48	TL	A	4	2

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST _____ AT VENDOR _____ IN FIELD

COMMENTS:

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 0230	E G	* *	DFASTHIC LAB TXT M48	TL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		LABEL

PART NUMBER RELATED CHANGES:

Part	Fr/To	Description	CC	St
3493 0230-003	E G	DFASTHIC LAB TXT M48	C	I

	From	To
ADD Desc:		DFASTHIC LAB TXT M48
CHG Status:	A	I
ADD Newpn:		3493 0230-004

Part	Fr/To	Description	CC	St
3493 0230-004	G	DFASTHIC LAB TXT M48	N	A

	From	To
ADD Desc:		DFASTHIC LAB TXT M48
ADD Status:		A

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 0230	E G	* *	DFASTHIC LAB TXT M48	TL	A	4	2

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST _____ AT VENDOR _____ IN FIELD

COMMENTS:

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 1014	E F	* *	DFASTQIC MSTR MED M17	TL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
ADD Med Cd:		CT
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description	CC	St
3493 1014-003	E F	DFASTQIC MSTR MED M17	C	I

	From	To
ADD Desc:		DFASTQIC MSTR MED M17
CHG Status:	A	I
ADD Newpn:		3493 1014-004

	Part Fr/To	Description	CC	St
3493 1014-004	F	DFASTQIC MSTR MED M17	N	A

	From	To
ADD Desc:		DFASTQIC MSTR MED M17
ADD Status:		A

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 1014	E F	* *	DFASTQIC MSTR MED M17	TL	A	4	2

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST _____ AT VENDOR _____ IN FIELD

COMMENTS:

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 1022	E F	* *	DFASTQIC LAB TXT M17	TL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
ADD Med Cd:		CT
CHG EPC:	*	COMMON
ADD PFC:		LABEL

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description	CC	St
3493 1022-003	E F	DFASTQIC LAB TXT M17	C	I

	From	To
ADD Desc:		DFASTQIC LAB TXT M17
CHG Status:	A	I
ADD Newpn:		3493 1022-004

	Part Fr/To	Description	CC	St
3493 1022-004	F	DFASTQIC LAB TXT M17	N	A

	From	To
ADD Desc:		DFASTQIC LAB TXT M17
ADD Status:		A

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 1022	E F	* *	DFASTQIC LAB TXT M17	TL	A	4	2

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST _____ AT VENDOR _____ IN FIELD

COMMENTS:

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 1030	E F	* *	DFASTMT MSTR MED M42	TL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description	CC	St
3493 1030-003	E F	DFASTMT MSTR MED M42	C	I

	From	To
ADD Desc:		DFASTMT MSTR MED M42
CHG Status:	A	I
ADD Newpn:		3493 1030-004

	Part Fr/To	Description	CC	St
3493 1030-004	F	DFASTMT MSTR MED M42	N	A

	From	To
ADD Desc:		DFASTMT MSTR MED M42
ADD Status:		A

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 1030	E F	* *	DFASTMT MSTR MED M42	TL	A	4	2

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST _____ AT VENDOR _____ IN FIELD

COMMENTS:

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 1048	E F	* *	DFASTMT LAB TXT M42	TL	A	4	1

CEDB CHANGES:
BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
ADD Med Cd:		MT
CHG EPC:	*	COMMON
ADD PFC:		LABEL

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description	CC	St
3493 1048-003	E F	DFASTMT LAB TXT M42	C	I

	From	To
ADD Desc:		DFASTMT LAB TXT M42
CHG Status:	A	I
ADD Newpn:		3493 1048-004

	Part Fr/To	Description	CC	St
3493 1048-004	F	DFASTMT LAB TXT M42	N	A

	From	To
ADD Desc:		DFASTMT LAB TXT M42
ADD Status:		A

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 1048	E F	* *	DFASTMT LAB TXT M42	TL	A	4	2

MANUFACTURING STATUS INFORMATION

DISPOSITION OF PARTS:

_____ SCRAP OLD PARTS _____ DEPLETE OLD STOCK _____ REWORK
_____ INCORP ON OPEN ORDERS _____ DRAW MATL FOR EVALUATN
_____ USE UNTIL NEW PARTS AVAIL _____ INCORP NEXT TIME BLD _____ OTHER

IMPLEMENT ON ASSEMBLIES/PARTS:

_____ SERIAL NUMBER: _____
_____ REWORK: _____ IN MFG _____ IN TEST _____ AT VENDOR _____ IN FIELD

COMMENTS:

TOP UNITS AFFECTED:

Top Unit	Part	Sta	Bom	DCL	Description
2346	1809-999	A	SU	MV	KIT: PTDTOQUAL
3445	3068-000	I	SU	MV	KIT,SCSI FIRMWARE DOWNLOAD
3445	3068-001	I	SU	MV	KIT,SCSI FIRMWARE DOWNLOAD
3445	3068-002	I	SU	MV	KIT,SCSI FIRMWARE DOWNLOAD
3445	3068-003	A	SU	MV	KIT,SCSI FIRMWARE DOWNLOAD
4490	2542-000	A	SU	MV	KIT, FCN-USR4000-010 A-MT
4490	2567-000	A	SU	MV	INDEX, FCN USR4000-010

3492 4639-003 DFASTMT MED ASSY M42 DFASTMT MED ASSY M42

DIFFERENCES BETWEEN 3492 4639-003 REPLACED-INACTIVE BY 3492 4639-004

	Find	Ch	Alt	From	To
CHG	1			3937 6975-000	3937 7007-000
DEL Ref	R001			3493 1030-003	
CHG Ref	R002			3493 1048-003	3493 1030-004
ADD Ref	R003				3493 1048-004

3492 5149-003 DFASTQIC MED ASSY M17 DFASTQIC MED ASSY M17

DIFFERENCES BETWEEN 3492 5149-003 REPLACED-INACTIVE BY 3492 5149-004

	Find	Ch	Alt	From	To
DEL Ref	R001			3493 1014-003	
CHG Ref	R002			3493 1022-003	3493 1014-004
ADD Ref	R003				3493 1022-004

3492 5156-003 DFASTHIC MED ASSY M48 DFASTHIC MED ASSY M48 LV 1.00

DIFFERENCES BETWEEN 3492 5156-003 REPLACED-INACTIVE BY 3492 5156-004

	Find	Ch	Alt	From	To
DEL Ref	R001			3493 0222-003	
CHG Ref	R002			3493 0230-003	3493 0222-004
ADD Ref	R003				3493 0230-004

EXHIBIT

E-9

```
%$ VERSION 90.032.001                                                    00000092
$$ SET STACK LIMIT 100                                                   00010000
$$ RESET LIST XREF                                                       00011000
$$ SET VERSION 01.005.000                                                00012000
$$ SET OMIT                                                              00013000
% PATCHFOR DOWNLOAD IN MULTIPLE CHUNKS                                   00013500010040006
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00014000
%                 |                                                      00015000
%      Class      |                    Unisys                            00016000
%                 |                                                      00017000
%                 | This material is restricted and proprietary to the   00018000
%     #######     | Unisys Corporation and is not to be reproduced, shown,00019000
%    ##     ##    | or disclosed outside the Unisys Corporation.  Customer00020000
%   ##       ##   | Services Engineering restricted and proprietary data 00021000
%   ##            | is furnished solely for use by Unisys personnel in   00022000
%   ##            | servicing customer's equipment.                      00023000
%   ##            |                                                      00024000
%   ##            |                                                      00025000
%   ##       ##   | This document is the property of and shall be returned00026000
%    ##     ##    | to Unisys Corporation, One Unisys Place, Detroit, MI 00027000
%     ######      | 48232.                                               00028000
%                 |                                                      00029000
%                 |                                                      00030000
%    Material     | Copyright (C) 1994.                                  00031000
%                 |                                                      00032000
%                 |                                                      00033000
%                 |                                                      00034000
%                 |                                                      00035000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00036000
$$ PAGE                                                                  00037000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00038000
%                                                                        00039000
%                  T A R G E T / F W L O A D                             00040000
%                                                                        00041000
%                        Patch History                                   00042000
%                                                                        00043000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00044000
  4/94  Initial version.  Will download A-code files to SBC controllers  00045000
        and SCSI Disk Drives.                                            00046000
  4/94  Changes - Qualification phase for official CSPO release 01.001.  00047000010010001
  5/94  Add 'Express Mode' for engineering (DISKs only) release 01.002.  00047200010020003
  6/94  Change location of some display statements. --- release 01.002.  00047500010020002
  10/94 Fixed seg array error in Verifyfile procedure.  release 01.002.  00047600010020004
                                                                         00048000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00049000
          DOWNLOAD FIRMWARE to A-SERIES TARGETS (DFAST)                  00052000
          =============================================                  00053000
                                                                         00054000
    Utility's Part Number: 3492 4639                                     00055000
                                                                         00056000
    This utility's primary function is to load microcode to SCSI BUS     00057000
    CONTROLLERs (SBC) and/or SCSI disk drives, and must be marked as a   00058000
    PPed (Privileged Program) in order to operate.                       00059000
                                                                         00060000
    Interaction with the user is via prompts at either a terminal or     00061000
    an ODT.  Terminal prompts use regular I/O.  ODT inputs use the       00062000
    ACCEPT statement.  Otherwise the input rules are the same.  The      00063000
    program requires interactive user participation to execute properly.00064000
                                                                         00065000
                                                                         00066000
Load:                                                                    00067000
-----                                                                    00068000
                                                                         00069000
    To operate, two basic elements must be present: an SBC controller or00070000
SCSI disk drive, and a microcode file on disk or tape.  The microcode    00071000
must reside on a unit served by a different controller or disk than the  00072000
one being initialized.  If a "critical unit" exists on the string served00073000
by the controller being initialized and there is only one path to that   00074000
critical unit, the system will reject the attempt to download the        00075000
microcode.                                                               00076000
                                                                         00077000
```

D1

```
    1) Step one is to determine the code file's capability by having the00078000
user enter a file name.  Normal Family Substitution rules are in        00079000
effect.  If the file cannot be found, the user is prompted to enter     00080000
another file name.  The file name may include an "ON <family>" in the   00081000
file declaration if it resides on disk (e.g., (UCODE)XYZ/123 ON MYPACK).00082000
If the file resides on tape, the file name only would be entered        00083000
(e.g., SCZFRM).                                                         00084000
                                                                        00085000
    If the code file is not a valid SBC or disk drive Acode file, the   00086000
file is rejected and the user is prompted to enter another file name.   00087000
                                                                        00088000
    2) Step two is to determine the SBC controller or drive capability. 00089000
The user enters a SBC or drive number and the utility reads the unit's  00090000
attributes.  If the SBC or drive number does not represent the correct  00091000
target, or if the target attributes do not match those of the Acode     00092000
file, the user is propted to enter a new target number or 'Quit'.  The  00093000
SBC controller or drive must be reserved.                               00094000
                                                                        00095000
>>>> There may be no way to prove that the SBC controller controls any  00096000
>>>>    given drive or that it is the only path to the drive or that    00097000
>>>>    another path exists!  The User should do an "OL' on the ODT     00098000
>>>>    to verify the paths available for the target.                   00099000
                                                                        00100000
                                                                        00101000
Verifyfile:                                                             00102000010020003
-----------                                                             00103000010020003
                                                                        00104000
    Verify is simpler than Load since no SBC or drive is involved.  The 00105000
user answers the file name prompt and the Verify routine used by Load is00106000
called to generate a report on the attributes associated with that file.00107000
                                                                        00108000
    Verify allows the user to cycle through multiple, potential A-code  00109000
files until 'Quit' is entered.                                          00110000
$$ POP OMIT                                                             00111000
BEGIN                                                                   00112000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00113000
%                        Structure Generating Declarations              00114000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00115000
DEFINE                                                                  00116000
  OVHD                  = 12   #        % At front of each code segment 00117000
 ,XSTATBYTES            = 254  #        % Read Unit Status return length00118000
 ,EIGHTK                = 8192 #                                        00119000
 ,ENDSGD                = #                                             00120000
 ,SYSCAP                = 393216 #                                      00120500010040006
 ,MAXELEMENTS           = 1 #     % MAX ELEMENT PER DIMENSION           0012060001.005.000
 ,MAXROWS               = 48  #   % MAX NUMBER OF ROWS                  0012080001.005.000
 ;                                                                      00121000
                                                                        00122000
FILE                                                                    00123000
  CODE( KIND            = TAPE,         % UNKNOWN TYPE                 00124000
%%%%%   LABELKIND       = UNLABELLED,   % READ ONLY                    00125000
        FILEUSE         = IN,                                           00126000
        OPTIONAL        = TRUE,                                         00127000
        NEWFILE         = FALSE,                                        00128000
        DEPENDENTSPECS  = TRUE)                                         00129000
 ,LINE( KIND            = PRINTER,                                      00130000
        FILEUSE         = OUT,                                          00131000
        FRAMESIZE       = 8,                                            00132000
        MAXRECSIZE      = 132)                                          00133000
 ,RMT(  KIND            = REMOTE,                                       00134000
        FILEUSE         = IO,                                           00135000
        BLOCKSTRUCTURE  = EXTERNAL,                                     00136000
        FRAMESIZE       = 8,                                            00137000
        MAXRECSIZE      = 132)                                          00138000
 ;                                                                      00139000
                                                                        00140000
DIRECT EBCDIC ARRAY                                                     00141000
  IMLBUF2       [0:0,0:0]               % To be resized                00142000010040006
 ,IMLBUF3       [0:0,0:0]               % To be resized                0014220001.005.001
 ,IMLBUF        [0:0]                                                  00142500010040006
 ;                                                                      00143000
```

DIM. IFFER

2 DIMENSIONA BUFFER

```
                                                                                00144000
EBCDIC ARRAY                                                                    00145000
  CODEREC      [0:0]                    % CODE record                          00146000
 ,ETSTMP       [0:23]                   % Timestamp data                       00147000
 ,FCUSTID      [0:7]                    % FW file customer ID                  00148000
 ,FNEWFWLEVEL  [0:3]                    % FW file new firmware level           00149000
 ,FVENDORID    [0:7]                    % FW file vendor ID                     00150000
 ,OLDFWLEVEL   [0:3]                    % FW level before download             00150500010020003
 ,HDR          [0:0]                    % CODE record zero                      00151000
 ,HDPRESULT    [0:29]                                                          00152000
 ,INBUF        [0:131]                  % User input buffer                     00153000
 ,L            [0:131]                  % Formatting space                      00154000
 ,SCR2         [0:131]                  % Scratch 2                             00154500010040006
 ,SCR          [0:131]                  % Scratch space                         00155000
 ,TTL          [0:89]                   % CODE.TITLE                            00156000
 ,SHORTBUF     [0:XSTATBYTES]           % UMR Interface calls                   00157000
 ,ESHORTBUF    [0:XSTATBYTES]           % ASCII to EBCDIC conversion            00158000
 ,SAVEFWLVL    [0:7]                    % FW file data area of FW level        00159000
 ,OEM          [0:71]                   % DRIVE TYPE                           00159500010020004
 ;                                                                              00160000
                                                                                00161000
REAL ARRAY                                                                      00162000
  RCODE[0]       = CODEREC                                                      00163000
 ,RHDR[0]        = HDR                                                          00164000
 ,TSTMP[0]       = ETSTMP                                                       00165000
 ,X[0:0]                                                                        00165500010020004
 ;                                                                              00166000
                                                                                00167000
HEX ARRAY                                                                       00168000
  HSCR[0]  = SCR;                                                               00169000
                                                                                00170000
POINTER                                                                         00171000
  DUMMY                                 % Never used                          00172000
 ,IMLP                                  % To IMLBUF                           00173000
 ,IMLP2                                 % To IMLBUF2                          0017320001.005.000
 ,IMLP3                                 % To IMLBUF3                          0017340001.005.000
 ,PC                                    % To CODEREC                          00174000
 ,PL                                    % To L                                00175000
 ;                                                                              00176000
                                                                                00177000
PICTURE                                                                         00178000
  DATENTIME(N/99I99I99" @ "N:99I99);                                            00179000
                                                                                00180000
TRUTHSET                                                                        00181000
  NUMBERS ("0123456789")                                                        00181200010020004
 ,HEXDIGITS (NUMBERS OR "ABCDEF")                                               00181400010020004
 ,ENDER (" .");                                                                 00182000010020004
                                                                                00183000
TRANSLATETABLE                                                                  00184000
  UPCASER(EBCDIC TO EBCDIC,                                                     00185000
          "abcdefghijklmnopqrstuvwxyz" TO                                       00186000
          "ABCDEFGHIJKLMNOPQRSTUVWXYZ");                                        00187000
                                                                                00188000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00189000
%               Real and Integer Variables                                      00190000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00191000
REAL                                                                            00192000
        % These Reals are either bit buckets or other values which             00193000
        % would not survive integerization.                                    00194000
  FKAIW                         % FILEKIND Attribute Info Word                 00195000
 ,MRD                           % Maintenance Result Descriptor bits           00196000
 ,SYSTEMTYPE                    % A11, A14, A16, A18, A19 are valid            00197000010010001
 ,VAL                           % Value of hex to decimal conversion           00197500010020004
 ;                                                                              00198000
                                                                                00199000
INTEGER                                                                         00200000
  CODEEOF                       % CODE.LASTRECORD                              00201000
 ,CODEMRS                       % CODE.MAXRECSIZE                              00202000
 ,CTLUNIT                       % PCD unit number of CTL                       00203000
 ,FHDRBYTES                     % # bytes in header record                     00204000
 ,FCODEBYTES                    % Code size (# bytes) in firmware file         00205000
```

```
  ,FUNCTION                     % Operating mode for this run           00206000
  ,LL                           % Number of bytes remaining in PL       00207000
  ,SEGINX                       % Segment starting record index         00208000
  ,SEGMRS                       % Segment maxrecsize                    00209000
  ,SOFAR                        % Bytes moved to IMLBUF so far          00210000
  ,TIMER                        % Drop dead timer (intervals)           00211000
  ,NUMREPFWLVLS                 % Number of replaceable firmware levels 00212000
  ,DRIVEIDLOC                   % Array index to FW drive IDs           00213000
  ,NUMDRIVEID                   % Number of drive IDs valid for the code00214000
  ,OPTODO                       % Opcode parameter to UMR               00215000
  ,NUMBROFIOS                   % # of IO's to do for multi chunk dwnld 00215200010040006
  ,SIZEOFLSTIO                  % Size of last IO to do multi chunk dnld00215400010040006
  ,NROWS                        % Number of rows required for download  00215500010040006
  ,MROWS                        % Number of MOD rows as residual rows   0021560001.005.000
  ;                                                                     00216000010040006
                                                                        00217000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00218000
%                       Boolean Variables                               00219000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00220000
BOOLEAN                                                                 00221000
  CTLASSIGNED                   % CTL assigned                          00222000
  ,SDMASSIGNED                  % SCSI disk drive assigned              00223000
  ,DEBUG                        % Internal debug mode                   00224000
  ,ODTMODE                      % User is at a console or terminal      00225000
  ,ATODT                        % User is at console                    00226000
  ,SKIPODTPRINT                 % Don't print ODT display               00226500010010001
  ,EXPRESSMODE                  % Omit many displays (for engineering)  00226600010020003
  ,VERIFYONLY                   % Don't build FW image in buffer        00226800010020004
  ,FIRSTTIME                    %                                       0022690001.005.000
  ;                                                                     00227000
                                                                        00228000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00229000
%                   Structure Independent Defines                       00230000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00231000
DEFINE                                                                  00232000
  ADD                       = REPLACE PL:PL BY # ;                      00233000
  ,B                        = BOOLEAN #;                                00234000
  ,BLANK                    = REPLACE L BY " " FOR 22 WORDS #;          00235000
  ,C                        = , #;                                      00236000
  ,CLEARINBUF               = REPLACE INBUF BY " " FOR 12 WORDS #;      00237000
  ,NUL                      = 48"00" #;                                 00238000
  ,P                        = POINTER #;                                00239000
  ,PUT(N)                   = REPLACE PL:L[N] BY #;                     00240000
  ,SHOW(S,TF)               = BEGIN                                     00241000
                                START S;                                00242000
                                SHOWIT(TF);                             00243000
                              END #;                                    00244000
  ,START                    = REPLACE PL:L BY #;                        00245000
                                                                        00246000
% Formatting constants                                                  00247000
                                                                        00248000
  ,TXTLOC30                   = 30 #;                                   00249000
  ,TXTLOC22                 = 22 #;                                     00250000
  ,TXTLOC15                 = 15 #;                                     00251000
  ,ERRLOC                   = (TXTLOC30+2) #;                           00252000
                                                                        00253000
% Function values                                                       00254000
                                                                        00255000
  ,LOADCTLV                 = 1 #;                                      00256000
  ,LOADEVV                  = 2 #;                                      0025700001.005.000
  ,VERIFYV                  = 3 #;                                      00258000
                                                                        00259000
% Code File Header (most recent version)                                00260000
%  Elements for all fields include their location and their length.     00261000
                                                                        00262000
  ,CUSTIDLOC                =  0 #;                                     00263000
  ,CUSTIDLNG                =  8 #;                                     00264000
%,CUSTIDEXP                 = "FUJITSU " #;                             00264500010040006
  ,CUSTIDEXP                = "UNISYS  " #;                             00265000010040006
  ,HDRECSIZELOC             =  8 #;                                     00266000
  ,HDRECSIZELNG             =  8 #;                                     00267000
```

```
  ,FWCODESIZELOC          = 16 #                                          00268000
  ,FWCODESIZELNG          =  8 #                                          00269000
  ,NEWFWLEVELOC           = 24 #                                          00270000
  ,NEWFWLEVELNG           =  4 #                                          00271000
  ,VENDORIDLOC            = 28 #                                          00272000
  ,VENDORIDLNG            =  8 #                                          00273000
  ,NUMDRIVEIDLOC          = 36 #                                          00274000
  ,NUMDRIVEIDLNG          =  4 #                                          00275000
  ,NUMREPFWLVLSLOC        = 40 #                                          00276000
  ,NUMREPFWLVLSLNG        =  4 #                                          00277000
  ,OLDFWLEVELNG           =  8 #                                          00278000
  ,DRIVEIDLNG             = 16 #                                          00279000
%,VENDORIDEXP             = "FUJITSU " #                                  00279500010040006
  ,VENDORIDEXP            = "UNISYS  " #                                  00280000010040006
%*,NOTESTRING              = "SBCFirmware."  #                             00281000
  ,TVENDORID              = ESHORTBUF [10] #    % FOR 8                   00282000
  ,TPRODID                = ESHORTBUF [18] #    % FOR 16                  00283000
  ,TPRODREVLVL            = ESHORTBUF [34] #    % FOR 4                   00284000
  ,TOEMID                 = ESHORTBUF [54] #    % FOR 8                   00284500010020004
  ,TSERVOREVLVL           = ESHORTBUF [14] #    % FOR 8                   0028460001.005.000
  ;                                                                       00285000
                                                                          00286000
% User Maint Requests                                                    00287000
                                                                          00288000
DEFINE                                                                    00289000
  ASSIGNUNITV             = 0 #                                           00290000
  ,UNASSIGNUNITV          = 1 #                                           00291000
  ,LOADSLAVEIMLV          = 2 #                                           00292000
  ,READERRORLOGV          = 3 #                                           00293000
  ,ATTRIBUTESV            = 4 #                                           00294000
  ,ASSIGNCTLV             = 5 #                                           00295000
  ,UNASSIGNCTLV           = 6 #                                           00296000
  ,MODESENSE              = 7 #                                           00296200010040006
  ,DOWNLOADMODE4          = 8 #                                           00296400010040006
  ,DOWNLOADMODE6          = 9 #                                           00296600010040006
  ,DOWNLOADMODE7          = 10 #                                          00296800010040006
  ,TESTUNITREADY          = 11 #                                          00296900010040006
  ,INQUIRY                = 12 #                                          0029695001.005.001
  ;                                                                       00297000
                                                                          00298000
LIBRARY                                                                   00299000
  MCP(  FUNCTIONNAME      = "MCPSUPPORT.",                                00300000
        LIBACCESS         = BYFUNCTION);                                  00301000
                                                                          00302000
INTEGER PROCEDURE USERMAINTREQUEST(                                       00303000
        UNITNUMBER, FUNCTIONCODE, IOLENGTH, BUFFEROFFSET, BUFFERID,       00304000
        BUFFER, MRD, DEVICERD);                                           00305000
VALUE   UNITNUMBER, FUNCTIONCODE, IOLENGTH, BUFFEROFFSET, BUFFERID;       00306000
INTEGER UNITNUMBER, FUNCTIONCODE, IOLENGTH, BUFFEROFFSET, BUFFERID, MRD;  00307000
EBCDIC ARRAY BUFFER[*], DEVICERD[*];                                      00308000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00309000
%. Description:                                                           00310000
%    This procedure is a Library-exported interface that supports        00311000
%    SCSI and IPI-3 I/O utilities (e.g., microcode download utility).     00312000
%. Parameters:                                                            00313000
%    UNITNUMBER          External unit number of the device being        00314000
%*                        accessed.  The MCP will determine whether a     00315000
%*                        port, CTL, or a device is being accessed using  00316000
%*                        data from the I/O unit tables.                  00317000
%    FUNCTIONCODE:       Case index to be executed in the interface.     00318000
%                        0: Assign Unit                                  00319000
%                        1: Unassign Unit                                00320000
%                        2: Do FW Load (Load Slave IML)                  00321000
%                        3: Read Extended Status (Read Error Log)        00322000
%                        4: Obtain Disk ID (Report Attributes)           00323000
%*                       5: Assign CTL                                   00324000
%*                       6: Unassign CTL                                 00325000
%    IOLENGTH            Bytes                                           00326000
%    BUFFEROFFSET        Used by SCSI interface to allow non-            00327000
%                        transmittable data to precede actual data       00328000
%                        in the buffer.                                  00329000
```

```
%*                      Not required for IPI.                                   00330000
%    BUFFERID           Not used by SCSI.                                       00331000
%*                      Not required for IPI.                                   00332000
%    BUFFER             Direct EBCDIC array to be used as the source            00333000
%                       of the data and the repository for extended             00334000
%                       parameter handling.                                     00335000
%    MRD                Maintenance Result Descriptor bits. Some bits           00336000
%                       refer to HDPRESULT as returned to PTD.                  00337000
%    DEVICERD           Status and Sense data                                   00338000
%. Assumptions:                                                                 00339000
%    This routine protects the integrity of the system from a poorly            00340000
%    written application-level calling program.                                 00341000
%. Results:                                                                     00342000
%       0:              Function succeeded.                                     00343000
%      >0:              Function failed.  See SHOWRSLT for details.             00344000
%. Locks:                                                                       00345000
%    None.                                                                      00346000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00347000
  LIBRARY MCP;                                                                  00348000
                                                                                00349000
LABEL                                                                           00350000
  MAINLOOP                                                                      00351000
 ,GRANDXIT;                                                                     00352000
$$ PAGE BEGINSEGMENT                                                            00353000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00353020010020004
%               CONVERT HEX TO DECIMAL                                          %00353040010020004
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00353060010020004
PROCEDURE CONVERTHEXTODEC;                                                      00353080010020004
  BEGIN                                                                         00353100010020004
                                                                                00353120010020004
   INTEGER I;                                                                   00353140010020004
   POINTER PTR;                                                                 00353160010020004
                                                                                00353162010020004
   REAL                                                                         00353164010020004
    TEMP                                                                        00353166010020004
    ,VAL;                                                                       00353168010020004
                                                                                00353180010020004
   VAL := 0;                                                                    00353200010020004
   PTR :=  SCR;                                                                 00353220010020004
                                                                                00353280010020004
   FOR I := 1 STEP -1 UNTIL 0 DO                                                00353300010020004
     BEGIN                                                                      00353320010020004
     IF PTR IN NUMBERS THEN                                                     00353340010020004
       TEMP := INTEGER(PTR,1) * 16**I                                           00353360010020004
     ELSE                                                                       00353380010020004
       CASE REAL(PTR,1) OF                                                      00353400010020004
         BEGIN                                                                  00353420010020004
           "A" :  TEMP := 10 * 16**I;                                           00353440010020004
           "B" :  TEMP := 11 * 16**I;                                           00353460010020004
           "C" :  TEMP := 12 * 16**I;                                           00353480010020004
           "D" :  TEMP := 13 * 16**I;                                           00353500010020004
           "E" :  TEMP := 14 * 16**I;                                           00353520010020004
           "F" :  TEMP := 15 * 16**I;                                           00353540010020004
          ELSE ;                                                                00353560010020004
         END;                                                                   00353580010020004
                                                                                00353600010020004
     PTR       := * + 1;                                                        00353620010020004
     VAL       := * + TEMP;                                                     00353640010020004
     END;                                                                       00353660010020004
                                                                                00353680010020004
REPLACE OEM BY " " FOR 12 WORDS;                                                00353685010020004
REPLACE OEM BY VAL FOR * DIGITS;                                                00353690010020004
                                                                                00353695010020004
END;  % PROCEDURE CONVERTHEXTODEC                                               00353700010020004
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00354000
%               Common I/O Routines Block                                       00355000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00356000
PROCEDURE SHOWIT(DSP);                                                          00357000
  VALUE          DSP;                                                           00358000
  BOOLEAN        DSP;                                                           00359000
%-------------------------------------------------------------------------------00360000
```

D6

```
% Purpose:                                                              00361000
%  Show L on LINE.                                                      00362000
%                                                                       00363000
% Parameters:                                                           00364000
%  DSP          True: Caller wants a copy sent to user.                 00365000
%                                                                       00366000
% Globals:                                                              00367000
%  ODTMODE      True:  ODT is normal location of user.                  00368000
%               False: Terminal is.                                     00369000
%                                                                       00370000
% Assumptions:                                                          00371000
%  L has message text.                                                  00372000
%  PL points to next place to add text.                                 00373000
%  L will be blanked out upon exit.                                     00374000
%----------------------------------------------------------------------00375000
BEGIN                                                                   00376000
  DEFINE SEGMENT = #;                                                   00377000
                                                                        00378000
  IF NOT SKIPODTPRINT THEN                                              00378500010010001
    WRITE(LINE,OFFSET(PL),L);          % Log the transaction           00379000010010001
                                                                        00380000
  IF DSP THEN                          % Send a copy to the user       00381000
    IF ODTMODE THEN                    %  at the console              00382000
    BEGIN                                                               00383000
      REPLACE PL BY NUL;                                                00384000
      DISPLAY(L);                                                       00385000
      WAIT((3));                       % Slow down ODT traffic         00386000
    END                                                                 00387000
    ELSE                               %  at his terminal              00388000
      WRITE(RMT,OFFSET(PL),L);                                          00389000
                                                                        00390000
  BLANK;                                                                00391000
                                                                        00392000
END; % Show It                                                          00393000
                                                                        00394000
PROCEDURE PROMPT;                                                       00395000
%----------------------------------------------------------------------00396000
% Purpose:                                                              00397000
%  Display text to the user and expect a response.                      00398000
%                                                                       00399000
% Globals:                                                              00400000
%  ODTMODE      How we are to perform this action.                      00401000
%                                                                       00402000
% Assumptions:                                                          00403000
%  L contains the output message.                                       00404000
%  PL points to the next valid character in L.                          00405000
%  L needs to contain the input response.                               00406000
%  Both the output and input will be logged.                            00407000
%  The output will be upcased after logging in its original form.       00408000
%  The output will be prescanned and deblanked using PL and LL.         00409000
%  Under NO condition should PROMPT call SHOWIT.  SHOWIT blanks out     00410000
%   L as it exits, thus destroying all of our user input.               00411000
%----------------------------------------------------------------------00412000
BEGIN                                                                   00413000
  BOOLEAN BRSLT;                                                        00414000
                                                                        00415000
  IF ODTMODE THEN                                                       00416000
  BEGIN                                                                 00417000
    WRITE(LINE,OFFSET(PL),L);          % Log it                        00418000
    REPLACE PL BY NUL;                                                  00419000
    ACCEPT(L);                         % Both output and input space   00420000
    SCAN L FOR LL:132 UNTIL = NUL;                                      00421000
    LL := 132 - LL;                                                     00422000
    WRITE(LINE,LL,L);                  % Log it again                  00423000
  END                                                                   00424000
  ELSE % Terminal mode                                                  00425000
  BEGIN                                                                 00426000
    WRITE(LINE,OFFSET(PL),L);          % Log it                        00427000
    WRITE(RMT, OFFSET(PL),L);          % Tell user                     00428000
    BRSLT := READ(RMT,80,L);                                            00429000
    IF BRSLT THEN                      % Error, maybe even ?END        00430000
```

```
      GO GRANDXIT;                                                          00431000
    PL := L[LL:=REAL(BRSLT).[47:20]];                                       00432000
    WRITE(LINE,LL,L);                    % Log it again                    00433000
  END;                                                                      00434000
                                                                            00435000
  REPLACE L BY L FOR LL WITH UPCASER;                                       00436000
  SCAN PL:L FOR LL:LL WHILE = " ";                                          00437000
END; % Prompt                                                               00438000
$$ ENDSEGMENT                                                               00439000
$$ PAGE BEGINSEGMENT                                                        00440000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00441000
%                  Debugging Module                                         00442000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00443000
% Contents:                                                                 00444000
%  SHOWMRDBITS                                                              00445000
%  SHOWHDPRESULT                                                            00446000
%  SHOWRSLT                                                                 00447000
%  SHOWATTRIBUTES                                                           00448000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00449000
PROCEDURE SHOWMRDBITS;                                                      00450000
%-------------------------------------------------------------------------00451000
% Purpose:                                                                 00452000
%  Expand the Maintenance Result Descriptor bits following a call to       00453000
%   USERMAINTREQUEST.                                                       00454000
%                                                                           00455000
% Globals:                                                                 00456000
%  MRD          The Maintenance Result Descriptor                          00457000
%                                                                           00458000
% Assumptions:                                                             00459000
%  The MRD is a packed bit field.                                           00460000
%  We don't know how to decode this yet.                                    00461000
%  This entire procedure will be replaced with defines, case statements,    00462000
%   value arrays, etc.                                                      00463000
%  For the moment, just dump the MRD in hex.                                00464000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00465000
BEGIN                                                                       00466000
  DEFINE SEGMENT = #;                                                       00467000
                                                                            00468000
  REPLACE SCR BY MRD FOR 6;                                                 00469000
  PUT(TXTLOC30)                                                             00470000
    "MRD = ",                                                               00471000
    HSCR FOR 12 WITH HEXTOEBCDIC;                                           00472000
  SHOWIT(TRUE);                                                             00473000010040006
                                                                            00474000
END; % Show MRD bits                                                        00475000
                                                                            00476000
PROCEDURE SHOWHDPRESULT;                                                    00477000
%-------------------------------------------------------------------------00478000
% Purpose:                                                                 00479000
%  Expand the HDPRESULT coming out of the USERMAINTREQUEST.                 00480000
%                                                                           00481000
% Globals:                                                                 00482000
%  HDPRESULT    A byte array containing RMM data.                          00483000
%                                                                           00484000
% Assumptions:                                                             00485000
%  We think that only 12 bytes of data are valid.  We may have              00486000
%   gotten an interface error if our array is too short.                    00487000
%  We don't know the details of what is in this thing yet.                  00488000
%  For now, we will just format the first 12 bytes in hex.                  00489000
%-------------------------------------------------------------------------00490000
BEGIN                                                                       00491000
                                                                            00492000
  REPLACE SCR BY HDPRESULT FOR 12;                                          00493000
  PUT(TXTLOC30)                                                             00494000
    "HDP = ",                                                               00495000
    HSCR[00] FOR 12 WITH HEXTOEBCDIC,                                       00496000
    " ",                                                                    00497000
    HSCR[12] FOR 16 WITH HEXTOEBCDIC;                                       0049800001.005.000
  SHOWIT(TRUE);                                                             00499000010040006
                                                                            00500000
END; % Show HDP result                                                      00501000
```

```
                                                                        00502000
PROCEDURE SHOWRSLT(RSLT,FCN);                                           00503000
  VALUE            RSLT,FCN;                                          00504000
  INTEGER          RSLT,FCN;                                          00505000
%-----------------------------------------------------------------------00506000
% Purpose:                                                             00507000
%  Expand the integer value contained in RSLT from a USERMAINTREQUEST   00508000
%   into understandable text.                                         00509000
%                                                                     00510000
% Parameters:                                                         00511000
%  RSLT          The procedure value returned from USERMAINTREQUEST.  00512000
%  FCN > 0       What function were we trying to do?                  00513000
%  FCN < 0       Elapsed time in Read Attributes.                     00514000
%                                                                     00515000
% Assumptions:                                                        00516000
%  If RSLT = 0, the result was clean.                                 00517000
%  If RSLT > 0, it is a CASE code for the actual error.               00518000
%       We can use this to make global decisions in addition to       00519000
%       just printing out the value.                                  00520000
%-----------------------------------------------------------------------00521000
BEGIN                                                                   00522000
  INTEGER MM,SS;                                                        00523000
  LABEL XIT;                                                            00524000
                                                                        00525000
  BLANK;                                  % Often follows a PROMPT       00526000
  IF FCN < 0 THEN                         % Format elapsed time          00527000
  BEGIN                                                                 00528000
    FCN := ABS(FCN);                                                    00529000
    MM  := FCN DIV 60;                                                  00530000
    SS  := FCN MOD 60;                                                  00531000
    START                                                               00532000
      MM FOR 2 DIGITS, ":",                                             00533000
%*    SS FOR 2 DIGITS, " Read Attributes: ",                            00534000
      SS FOR 2 DIGITS, " Target: ",                                     00535000
      CTLUNIT FOR * DIGITS,                                             00536000
      " storing firmware.";                                             00537000
    GO XIT;                                                             00538000
  END                                                                   00539000
  ELSE                                                                  00540000
  CASE FCN OF                                                           00541000
  BEGIN                                                                 00542000
ASSIGNUNITV:                                                            00543000
    START "Assign Unit: ",      CTLUNIT FOR * DIGITS;                   00544000
UNASSIGNUNITV:                                                          00545000
    START "Unassign Unit: ",    CTLUNIT FOR * DIGITS;                   00546000
LOADSLAVEIMLV:                                                          00547000
    START "Load Slave FW:  ",   CTLUNIT FOR * DIGITS;                   00548000
READERRORLOGV:                                                          00549000
    START "Read Error Log: ",   CTLUNIT FOR * DIGITS;                   00550000
ATTRIBUTESV:                                                            00551000
    START "Read Attributes: ", CTLUNIT FOR * DIGITS;                    00552000
ASSIGNCTLV:                                                             00553000
    START "Assign CTL: ",       CTLUNIT FOR * DIGITS;                   00554000
UNASSIGNCTLV:                                                           00555000
    START "Unassign CTL: ",     CTLUNIT FOR * DIGITS;                   00556000
  END;                                                                  00557000
                                                                        00558000
  PUT(TXTLOC22)                                                         00559000
    "RSLT = ",                                                          00560000
    RSLT FOR * DIGITS,                                                  00561000
    ": ";                                                               00562000
  CASE RSLT OF                                                          00563000
  BEGIN                                                                 00564000
  0: ADD "No error";                                                    00565000
  1: ADD "Unit not supported (Vendor ID not UNISYS)";                   00566000
  3: ADD "Unit is not reserved (URed)";                                 00567000
  5: ADD "Illegal/undefined function code specified";                   00568000
  7: ADD "I/O error. See MRD & DEVICERD parameters";                    00569000
  9: ADD "I/O length exceeds buffer's capacity";                        00570000
 11: ADD "UMR can't assign unit (reason unknown)";                      00571000
 13: ADD "Target is marked 'broken' by MCP";                            00572000
```

```
  15: ADD "Target is marked 'in use' by MCP";                                  00573000
  17: ADD "Invalid target number was specified";                               00574000
  19: ADD "Unit not owned (FREE) by this partition";                           00575000
  21: ADD "I/O cannot be initiated";                                           00576000
  23: ADD "UMR Can't determine target 'unit type'";                            00577000
  25: ADD "UMR issued I/O to unassigned target";                               00578000
  27: ADD "Unit not present (FREEd)";                                          00579000
  29: ADD "Can't find usable path to target";                                  00580000
  31: ADD "Illegal Unassign";                                                  00580200010040006
  33: ADD "IO length exceeds Max IO of 393216";                                00580400010040006
  ELSE: ;                                                                      00581000
   END;                                                                        00582000
                                                                               00583000
XIT:                                                                           00584000
  SHOWIT(TRUE);                                                                00585000
                                                                               00586000
END; % Show result                                                             00587000
                                                                               00588000
PROCEDURE SHOWATTRIBUTES;                                                      00589000
%----------------------------------------------------------------------------00590000
% Purpose:                                                                     00591000
%  Expand the text in the SHORTBUF following a call to USERMAINTREQUEST       00592000
%   with the ATTRIBUTESV parameter.                                           00593000
%                                                                              00594000
% Globals:                                                                     00595000
%  SHORTBUF     Contains the text results of ATTRIBUTES command.              00596000
%                                                                              00597000
%----------------------------------------------------------------------------00598000
BEGIN                                                                          00599000
                                                                               00600000
  REPLACE  SCR [0] BY " " FOR 2;                                               00600100010020004
  REPLACE  POINTER(X[0]) BY  SHORTBUF [54] FOR 1;                              00600200010020004
  REPLACE SCR[0] BY POINTER(X[0],4) FOR 2 WITH HEXTOEBCDIC;                    00600250010020004
% REPLACE  SCR BY "C8";                                                        00600300010020004
% SHOW ("  SCR = " C  SCR FOR 2, TRUE);                                       00600400010020004
                                                                               00600500010020004
  CONVERTHEXTODEC;                                                             00600600010020004
                                                                               00600700010020004
  REPLACE ESHORTBUF [0] BY SHORTBUF [0] FOR SIZE (ESHORTBUF)                   00601000
        WITH ASCIITOEBCDIC;                                                    00602000
  IF EXPRESSMODE THEN    % Don't display - only print out                      00602100010020003
    BEGIN                                                                      00602150010020003
    SHOW ("TARGET Attributes: "          C                                     00602200010020003
          "VENDOR ID - "                 C                                     00602250010020003
          TVENDORID   FOR VENDORIDLNG    C  ".", FALSE);                       00602300010020003
    SHOW ("PRODUCT ID - "                C TPRODID FOR DRIVEIDLNG      C 00602350010020004
          "- "                           C OEM FOR 3 C ".", FALSE);            00602400010020004
    SHOW ("FW LEVEL - "                  C                                     00602450010020003
          TPRODREVLVL FOR NEWFWLEVELNG   C  ".", FALSE);                       00602500010020003
    END                                                                        00602550010020003
  ELSE                  % Display and print                                    00602600010020003
    BEGIN                                                                      00602650010020003
    SHOW ("TARGET Attributes: "          C                                     00603000010020003
          "VENDOR ID - "                 C                                     00604000010020003
          TVENDORID   FOR VENDORIDLNG    C  ".", TRUE);                        00605000010020003
    SHOW ("PRODUCT ID - "                C TPRODID FOR DRIVEIDLNG      C 00606000010020004
          "- "                           C OEM FOR 3 C ".", TRUE);             00607000010020004
    SHOW ("FW LEVEL - "                  C                                     00608000010020003
          TPRODREVLVL FOR NEWFWLEVELNG   C  ".", TRUE);                        00609000010020003
    END;                                                                       00609500010020003
                                                                               00610000
END; % Show attributes                                                         00611000
                                                                               00612000
PROCEDURE SHOWINQUIRYDATA;                                                     0061202001.005.000
%----------------------------------------------------------------------------0061204001.005.000
% Purpose:                                                                     0061206001.005.000
%  Expand the text in the SHORTBUF following a call to USERMAINTREQUEST       0061208001.005.000
%   with the INQUIRY  parameter.                                              0061210001.005.000
%                                                                              0061212001.005.000
% Globals:                                                                     0061214001.005.000
%  SHORTBUF     Contains the text results of ATTRIBUTES command.              0061216001.005.000
```

```
%                                                                          0061218001.005.000
%-------------------------------------------------------------------------0061220001.005.000
BEGIN                                                                      0061222001.005.000
                                                                           0061224001.005.000
  REPLACE  SCR [0] BY " " FOR 2;                                           0061226001.005.000
  REPLACE  POINTER(X[0]) BY  SHORTBUF [54] FOR 1;                          0061228001.005.000
  REPLACE SCR[0] BY POINTER(X[0],4) FOR 2 WITH HEXTOEBCDIC;                0061230001.005.000
% REPLACE  SCR BY "C8";                                                    0061232001.005.000
% SHOW ("  SCR = " C  SCR FOR 2, TRUE);                                    0061234001.005.000
                                                                           0061236001.005.000
  CONVERTHEXTODEC;                                                         0061238001.005.000
                                                                           0061240001.005.000
  REPLACE ESHORTBUF [0] BY SHORTBUF [0] FOR SIZE (ESHORTBUF)               0061242001.005.000
        WITH ASCIITOEBCDIC;                                                0061244001.005.000
  IF EXPRESSMODE THEN    % Don't display - only print out                  0061246001.005.000
    BEGIN                                                                  0061248001.005.000
    SHOW ("TARGET Inqiry Data: ",FALSE);                                   0061250001.005.000
%   SHOW ("TARGET Attributes: "          C                                 0061252001.005.000
%        "VENDOR ID - "                  C                                 0061254001.005.000
%        TVENDORID   FOR VENDORIDLNG     C  ".", FALSE);                   0061256001.005.000
%   SHOW ("PRODUCT ID - "                C TPRODID FOR DRIVEIDLNG      C 0061258001.005.000
%        "- "                            C OEM FOR 3 C ".", FALSE);        0061260001.005.000
    SHOW ("SERVO FW LEVEL - "                  C                           0061262001.005.000
        TSERVOREVLVL FOR NEWFWLEVELNG  C  ".", FALSE);                     0061264001.005.000
    END                                                                    0061266001.005.000
  ELSE                  % Display and print                                0061268001.005.000
    BEGIN                                                                  0061270001.005.000
    SHOW ("TARGET Inquiry Data:",TRUE);                                    0061272001.005.000
%   SHOW ("TARGET Attributes: "          C                                 0061274001.005.000
%        "VENDOR ID - "                  C                                 0061276001.005.000
%        TVENDORID   FOR VENDORIDLNG     C  ".", TRUE);                    0061278001.005.000
%   SHOW ("PRODUCT ID - "                C TPRODID FOR DRIVEIDLNG      C 0061280001.005.000
%        "- "                            C OEM FOR 3 C ".", TRUE);         0061282001.005.000
    SHOW ("SERVO FW LEVEL - "                  C                           0061284001.005.000
        TSERVOREVLVL FOR NEWFWLEVELNG  C  ".", TRUE);                      0061286001.005.000
    END;                                                                   0061288001.005.000
                                                                           0061290001.005.000
END; % Show InquiryData                                                    0061292001.005.000
                                                                           0061294001.005.000
$$ ENDSEGMENT                                                              00613000
$$ PAGE                                                                    00614000
                                                                           00615000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00616000
%                Verification Block                                        00617000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00618000
                                                                           00619000
PROCEDURE READFCN;                                                         00619050010020002
BEGIN                                                                      00619100010020002
                                                                           00619150010020002
  SHOW ("                                                  ", TRUE);       00619200010020002
  SHOW (">>>>>>>>>>>>>>>>>>> ATTENTION <<<<<<<<<<<<<<<<<<<", TRUE);       00619250010020002
  IF NOT ATODT THEN    % AT A TERMINAL                                     00619300010020002
    BEGIN                                                                  00619350010020002
    SHOW (" READ the appropriate FCN for this firmware file   ", TRUE);    00619400010020002
    SHOW (" before downloading the firmware                   ", TRUE);    00619450010020003
    END                                                                    00619500010020002
  ELSE                 % AT AN ODT                                         00619550010020002
    BEGIN                                                                  00619600010020002
    SHOW (" before downloading the firmware                   ", TRUE);    00619650010020003
    SHOW (" READ the appropriate FCN for this firmware file   ", TRUE);    00619700010020002
    END;                                                                   00619750010020002
  SHOW (">>>>>>>>>>>>>>>>>>> ATTENTION <<<<<<<<<<<<<<<<<<<", TRUE);       00619800010020002
  SHOW ("                                                  ", TRUE);       00619850010020002
                                                                           00619900010020002
END;  % PROCEDURE READFCN                                                  00619950010020002
                                                                           00620000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00621000
%              PROCEDURE USERPROCESS                                       00622000010020002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00623000
                                                                           00624000
PROCEDURE USERPROCESS;                                                     00625000
```

```
BEGIN                                                                   00626000
LABEL LOOPER;                                                           00627000
                                                                        00628000
%*IF NOT ATODT THEN     % AT A TERMINAL                                 00629000
%*BEGIN                                                                 00630000
SHOW ("DFAST is a firmware download utility for SCSI disk drives",TRUE);00631000
SHOW ("and/or SBC Controllers.  DFAST requires:                ",TRUE);00632000
SHOW ("                                                        ",TRUE);00633000
SHOW ("   1.)  Firmware input file on disk or tape; e.g.,      ",TRUE);00634000
SHOW ("                                                        ",TRUE);00635000
SHOW ("       (UCODE)FWFILE/123 ON ANYPACK - if on disk        ",TRUE);00636000
SHOW ("                  or                                    ",TRUE);00637000
SHOW ("       (UCODE)FWFILE/123            - if on tape        ",TRUE);00638000
SHOW ("                                                        ",TRUE);00639000
SHOW ("   2.)  A target (SCSI disk drive or SBC Controller)    ",TRUE);00640000
SHOW ("        that has been reserved (URed).                  ",TRUE);00641000
SHOW ("                                                        ",TRUE);00642000
SHOW ("   3.)  '<Mix #> AX' when responding to a prompt.       ",TRUE);00643000
SHOW ("                                                        ",TRUE);00644000
IF NOT ATODT THEN                                                       00645000
   WAIT ((5));                                                          00646000
SHOW ("The process follows the following format:               ",TRUE);00647000
SHOW ("                                                        ",TRUE);00648000
SHOW ("A.) The user is asked to select one of four options:    ",TRUE);00649000
SHOW ("  1.) 'Ctlload'    - select a SBC unit type.            ",TRUE);00650000010020003
SHOW ("  2.) 'Devload'    - select a SCSI disk drive unit type. ",TRUE); 0065100001.005.000
SHOW ("  3.) 'Verifyfile' - check and display firmware file    ",TRUE);00652000010020003
SHOW ("                     header information without        ",TRUE);00653000010020003
SHOW ("                     performing a download to the target.",TRUE);00654000010020003
SHOW ("  4.) 'Quit'       - exit DFAST.                        ",TRUE);00655000010020003
SHOW ("B.) The user is prompted for a firmware file name or    ",TRUE);00656000
SHOW ("    'Quit' to exit DFAST.                               ",TRUE);00657000
SHOW ("C.) If the file is validated as a firmware file, the new ",TRUE);00658000
SHOW ("    firmware level is displayed; otherwise the program  ",TRUE);00659000
SHOW ("    returns to step 'B'.                                ",TRUE);00660000
IF NOT ATODT THEN                                                       00661000
   WAIT ((5));                                                          00662000
SHOW ("D.) If 'Ctlload' or 'Diskload' was entered in step 'A',  ",TRUE);00663000
SHOW ("    then for every SCSI disk or SBC unit number entered, ",TRUE);00664000
SHOW ("    steps 'a' through 'e' below are performed.          ",TRUE);00665000
SHOW ("a.) Target attributes are obtained and displayed.       ",TRUE);00666000
SHOW ("b.) If download is not allowed (due to mismatch of file/ ",TRUE);00667000
SHOW ("    target Firmware level, Product ID or Vendor ID), the ",TRUE);00668000
SHOW ("    process stops for that unit, and the user is asked  ",TRUE);00669000
SHOW ("    to enter another target number of the same unit type ",TRUE);00670000
SHOW ("    or 'End' to return to step 'A'.                     ",TRUE);00671000
SHOW ("    otherwise:                                          ",TRUE);00672000
SHOW ("c.) If the SCSI drive (not SBC) requires formatting(IVR) ",TRUE);00673000010020004
SHOW ("    after the download, the user is consulted to:       ",TRUE);00674000
SHOW ("  i.)  continue with the firmware download  -or-        ",TRUE);00675000
SHOW ("  ii.) stop the process for that unit (to back up data  ",TRUE);00676000
SHOW ("       on that drive) and select another target of the  ",TRUE);00677000
SHOW ("       same unit type, or 'End' to return to step 'A'.  ",TRUE);00678000
SHOW ("d.) After the download is complete and the target has   ",TRUE);00679000
SHOW ("    sequenced, the INQUIRY command is reissued, and the ",TRUE);00680000
SHOW ("    new firmware level of the target is displayed.      ",TRUE);00681000
SHOW ("e.) The user is prompted to enter another target number ",TRUE);00682000
SHOW ("    of the same unit type or 'End' to return to step 'A'.",TRUE);00683000
IF NOT ATODT THEN                                                       00684000
   WAIT ((5));                                                          00685000
%*END                                                                   00686000
$$ SET OMIT                                                             00687000
  ELSE                  % AT AN A-SERIES CONSOLE                        00688000
  BEGIN                                                                 00689000
    SHOW ("       QUIT the program.                          ", TRUE);  00690000
    SHOW ("       another device type or Firmware file or    ", TRUE);  00691000
    SHOW ("       or SBC) or exit to outer block to select   ", TRUE);  00692000
    SHOW ("   f.) Loop back for next unit of same type (disk ", TRUE);  00693000
    SHOW ("   e.) If error free, the process completes.      ", TRUE);  00694000
    SHOW ("       the process for that unit.                 ", TRUE);  00695000
    SHOW ("       either continue with downloading or stop   ", TRUE);  00696000
```

```
    SHOW ("       after download, the user is consulted to   ", TRUE);
    SHOW ("   d.) If the drive (not SBC) requires formatting ", TRUE);
    SHOW ("       (IVR) otherwise                            ", TRUE);
    SHOW ("       Vendor ID) the process stops for that unit.", TRUE);
    SHOW ("       of unit/file Firmware level, Product ID or ", TRUE);
    SHOW ("   c.) If download is not allowed (due to mismatch", TRUE);
    SHOW ("   b.) Unit information is obtained (INQUIRY).    ", TRUE);
    SHOW ("   a.) The unit must be RESERVED (URed).          ", TRUE);
    SHOW ("  for every Disk or SBC unit entered by the user: ", TRUE);
    SHOW ("  If the firmware file is read in properly, then  ", TRUE);
    SHOW ("The process follows the following format:         ", TRUE);
    SHOW ("                                                  ", TRUE);
    SHOW ("   2.)  SCSI Disk Drive or SBC Controller         ", TRUE);
    SHOW ("          e.g., (UCODE)FWFILE/123 ON ANYPACK      ", TRUE);
    SHOW ("   1.)  Firmware input file                       ", TRUE);
    SHOW (" drives and/or SBC Controllers.  DFAST requires:  ", TRUE);
    SHOW ("DFAST is a firmware download utility for SCSI disk", TRUE);
  END;
$$ POP OMIT

  READFCN;

LOOPER:
  START " Do you want to continue?  Enter 'YES' or 'NO'     ";
  PROMPT;
  IF PL = "N" THEN
    GO GRANDXIT
  ELSE
    IF PL NEQ "Y" THEN
      GO LOOPER;

END;  % BOOLEAN PROCEDURE USERPROCESS

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

BOOLEAN PROCEDURE OKTOFORMAT;
BEGIN

IF NOT EXPRESSMODE THEN
  BEGIN
  SHOW ("                                                  ", TRUE);

  IF NOT ATODT THEN      % AT A TERMINAL
    BEGIN
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    END
  ELSE                   % AT AN A-SERIES CONSOLE
    BEGIN
    SKIPODTPRINT := TRUE;   % Display to console only - don't print
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", TRUE);
    SKIPODTPRINT := FALSE;  % Print only - don't display to console
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", FALSE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", FALSE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", FALSE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", FALSE);
    SHOW (">>>WARNING, DRIVE NEEDS FORMAT(IVR) AFTER DOWNLOAD", FALSE);
    END;

  SHOW ("                                                  ", TRUE);
  SHOW ("                 USER BEWARE !!!!                 ", TRUE);
  SHOW ("                                                  ", TRUE);
END;   % IF NOT EXPRESSMODE
```

```
  SHOW ("Unit " C CTLUNIT FOR * DIGITS C " will need formatting " C      00770000010010001
        "(IVR) after firmware is loaded!", TRUE);                       00771000010020004
  SHOW ("                                                 ", TRUE);     00778000010010001
  SHOW ("DO YOU STILL WANT TO CONTINUE WITH THE DOWNLOAD?  ", TRUE);     00779000010010001
  START "Answer 'YES' or 'NO'";                                         00780000010010001
  PROMPT;                                                               00781000
  IF PL = "Y" THEN                                                      00782000
    OKTOFORMAT := TRUE                                                  00783000
  ELSE                                                                  00784000
    OKTOFORMAT := FALSE;                                                00785000
                                                                        00786000
END;  % PROCEDURE OKTOFORMAT                                            00787000
                                                                        00788000
BOOLEAN PROCEDURE VALIDATEACODE;                                        00789000
%-----------------------------------------------------------------------00790000
% Purpose:                                                              00791000
%  The user has received an unknown file from somewhere.                00792000
%  He wants to see if it is a valid SBC firmware file.                  00793000
%                                                                       00794000
% Procedure Result:                                                     00795000
%  True         The code file is valid.                                 00796000
%  False        Invalid file for the target.                            00797000
%                                                                       00798000
% Assumptions:                                                          00799000
%  We don't trust the caller.                                           00800000
%-----------------------------------------------------------------------00801000
BEGIN                                                                   00802000
  REAL X;                                                               00803000
  INTEGER FK;                                                           00804000
  BOOLEAN RSLT;                                                         00805000
  LABEL XIT;                                                            00806000
                                                                        00807000
  RSLT := TRUE;                          % Assume good file            00808000
                                                                        00809000
  SHOWIT(FALSE);                         % Blank line                  00810000
  IF EXPRESSMODE THEN                                                   00810200010020003
    SHOW("File Validation Report:",FALSE)                               00810400010020003
  ELSE                                                                  00810600010020003
    SHOW("File Validation Report:",TRUE);                               00811000010020003
                                                                        00813000
  START "Code file:";                                                   00814000
  PUT(TXTLOC15) CODE.TITLE;                                             00815000
  IF EXPRESSMODE THEN                                                   00815200010020003
    SHOWIT(FALSE)                                                       00815400010020003
  ELSE                                                                  00815600010020003
    SHOWIT(TRUE);                                                       00816000010020003
                                                                        00817000
  START "Customer ID:";                                                 00818000
  PUT(TXTLOC30)                                                         00819000
    ">",                                                                00820000
    HDR[CUSTIDLOC] FOR CUSTIDLNG,                                       00821000
    "<";                                                                00822000
  IF HDR[CUSTIDLOC] NEQ CUSTIDEXP THEN                                  00823000
  BEGIN                                                                 00824000
    SHOWIT(TRUE);                                                       00825000
    START "  Expecting:";                                               00826000
    PUT(TXTLOC30)                                                       00827000
      ">",                                                              00828000
      CUSTIDEXP,                                                        00829000
      "<";                                                              00830000
    SHOWIT(TRUE);                                                       00831000
    RSLT := FALSE;                                                      00832000
  END                                                                   00833000
  ELSE                                                                  00834000
    IF EXPRESSMODE THEN                                                 00834200010020003
      SHOWIT(FALSE)                                                     00834400010020003
    ELSE                                                                00834600010020003
      SHOWIT(TRUE);                                                     00835000010020003
                                                                        00836000
  START "Vendor ID:";                                                   00837000
  PUT(TXTLOC30)                                                         00838000
```

```
    ">",                                                                    00839000
    HDR[VENDORIDLOC] FOR VENDORIDLNG,                                       00840000
    "<";                                                                    00841000
  IF HDR[VENDORIDLOC] NEQ VENDORIDEXP THEN                                  00842000
  BEGIN                                                                     00843000
    SHOWIT(TRUE);                                                           00844000
    START "  Expecting:";                                                   00845000
    PUT(TXTLOC30)                                                           00846000
      ">",                                                                  00847000
      VENDORIDEXP,                                                          00848000
      "<";                                                                  00849000
    SHOWIT(TRUE);                                                           00850000
    RSLT := FALSE;                                                          00851000
  END                                                                       00852000
  ELSE                                                                      00853000
    IF EXPRESSMODE THEN                                                     00853200010020003
      SHOWIT(FALSE)                                                         00853400010020003
    ELSE                                                                    00853600010020003
      SHOWIT(TRUE);                                                         00854000010020003
                                                                            00855000
  START "New Firmware Level:";                                              00856000
  PUT(TXTLOC30)                                                             00857000
    ">",                                                                    00858000
    HDR[NEWFWLEVELOC] FOR NEWFWLEVELNG,                                     00859000
    "<";                                                                    00860000
  IF EXPRESSMODE THEN                                                       00860200010020003
    SHOWIT(FALSE)                                                           00860400010020003
  ELSE                                                                      00860600010020003
    SHOWIT(TRUE);                                                           00861000010020003
                                                                            00862000
                                                                            00863000
XIT:                                                                        00864000
  VALIDATEACODE := RSLT;                                                    00865000
  SHOWIT(FALSE);                        % Blank line                       00866000
END; % Validate Acode File                                                  00867000
                                                                            00868000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00869000
%          Compare Product IDs of Target and Input File                   %00870000010010001
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00871000
                                                                            00872000
BOOLEAN PROCEDURE COMPPRODIDS;                                              00873000010010001
BEGIN                                                                       00874000
  BOOLEAN   TARGETMATCH;                                                    00875000
  INTEGER   I, DRIVELOC;                                                    00876000
                                                                            00877000
  % Scan through valid Drive IDs in firmware file & match with Target       00885000
                                                                            00886000
  I := 0;                                                                   00887000
  DRIVELOC := DRIVEIDLOC;     % STARTING LOCATION IN HEADER RECORD           00888000
                                                                            00889000
  WHILE (((I:= *+1) LEQ NUMDRIVEID) AND (NOT TARGETMATCH)) DO               00890000
    BEGIN                                                                   00891000
    TARGETMATCH := TPRODID = HDR [DRIVELOC] FOR DRIVEIDLNG;                 00892000
    IF DEBUG THEN                                                           00893000
      SHOW ("FILE PRODID-LOOP = " C HDR [DRIVELOC] FOR DRIVEIDLNG,TRUE);    00894000
    DRIVELOC    := * + DRIVEIDLNG;                                          00895000
    END;                                                                    00896000
                                                                            00897000
  IF NOT TARGETMATCH THEN                                                   00898000
     BEGIN                                                                  00899000
     SHOW ("No match found in firmware file for Target " C                  00900000
           CTLUNIT FOR * DIGITS                           C                 00901000
           " with product ID = "                          C                 00902000
           TPRODID FOR DRIVEIDLNG,TRUE);                                    00903000
     SHOW ("<< Download will NOT take place. >>",TRUE);                     00904000
     COMPPRODIDS := FALSE;                                                  00905000010010001
     END                                                                    00906000
  ELSE                                                                      00907000
     COMPPRODIDS := TRUE;                                                   00908000010010001
                                                                            00909000
END;  % PROCEDURE COMPPRODIDS                                               00910000010010001
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 Store Firmware File Information
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PROCEDURE STOREFWFILEINFO;
BEGIN

INTEGER I;

 % Assign Code file information from header record to program variables.

  REPLACE FCUSTID      [0] BY HDR [CUSTIDLOC]     FOR CUSTIDLNG;
  REPLACE FNEWFWLEVEL [0] BY HDR [NEWFWLEVELOC] FOR NEWFWLEVELNG;
  REPLACE FVENDORID    [0] BY HDR [VENDORIDLOC]   FOR VENDORIDLNG;
  FHDRBYTES      := INTEGER (HDR [HDRECSIZELOC],     HDRECSIZELNG);
  FCODEBYTES     := INTEGER (HDR [FWCODESIZELOC],    FWCODESIZELNG);
  NUMDRIVEID     := INTEGER (HDR [NUMDRIVEIDLOC],    NUMDRIVEIDLNG);
  NUMREPFWLVLS := INTEGER (HDR [NUMREPFWLVLSLOC], NUMREPFWLVLSLNG);
  DRIVEIDLOC     := NUMREPFWLVLSLOC + NUMREPFWLVLSLNG +
                    (NUMREPFWLVLS * OLDFWLEVELNG);

  IF DEBUG THEN
     BEGIN
     SHOW ("FROM FILE -- "  C
        "CUSTID : "         C  FCUSTID      FOR CUSTIDLNG              C
        ". NEW FW LEVEL : " C  FNEWFWLEVEL  FOR NEWFWLEVELNG,TRUE);
     SHOW ("FROM FILE -- "  C
        "VENDORID : "       C  FVENDORID    FOR VENDORIDLNG            C
      ". HEADERBYTES : "    C  FHDRBYTES  FOR HDRECSIZELNG DIGITS,TRUE);
     SHOW ("FROM FILE -- "  C
      "CODEBYTES : "        C  FCODEBYTES   FOR FWCODESIZELNG DIGITS   C
      ". REPFWLVLS : " C  NUMREPFWLVLS FOR NUMREPFWLVLSLNG DIGITS,TRUE);
     SHOW ("FROM FILE -- "  C
      "NUMDRIVES : "        C  NUMDRIVEID   FOR NUMDRIVEIDLNG DIGITS   C
     ". DRIVEIDLOC: "       C  DRIVEIDLOC   FOR DRIVEIDLNG DIGITS,TRUE);
     END;

  IF NOT VERIFYONLY THEN

  BEGIN
  % Resize the IML buffer
%    SHOW("RESIZE THE IML BUFFER.",TRUE);
     IF FCODEBYTES >= SYSCAP THEN
       BEGIN
         NROWS := FCODEBYTES DIV EIGHTK;
%    REPLACE SCR[0] BY "NROWS= ", NROWS FOR * DIGITS;
%    WRITE(RMT,*//,SCR);
         IF NROWS >= MAXROWS THEN
           BEGIN
             RESIZE(IMLBUF2[*,*],MAXROWS,RETAIN);
             NROWS := MAXROWS;
             NROWS := * - 1;
             FOR I:= 0 STEP 1 UNTIL NROWS DO
             BEGIN
%              SHOW(" I = " C I FOR * DIGITS,TRUE);
               RESIZE (IMLBUF2[I,*],EIGHTK,RETAIN);
             END;
             IMLP2 := IMLBUF2[0,*];          % ASSIGN POINTER
             NROWS := (FCODEBYTES - SYSCAP) DIV EIGHTK;
             NROWS := * + 1;
             RESIZE(IMLBUF3[*,*],NROWS,RETAIN);
             NROWS := * - 1;
             FOR I:= 0 STEP 1 UNTIL NROWS DO
             BEGIN
%              SHOW(" I = " C I FOR * DIGITS,TRUE);
               RESIZE(IMLBUF3[I,*],EIGHTK,RETAIN);
             END;
             IMLP3 := IMLBUF3[0,*];
           END
       END
```

```
      ELSE                                                                  00951200010040006
        RESIZE(IMLBUF,FCODEBYTES,DISCARD);                                  00951400010040006
        IMLP := IMLBUF;                                                     00951600010040006
%       SHOW("ASSIGN POINTER TO RESIZED IMLBUF.",TRUE);                     0095180001.005.000
  % Build IML image from Code File                                          00952000
                                                                            00953000
  IF NOT EXPRESSMODE THEN                                                   00953500010020003
    SHOW("Building FW image in buffer.",TRUE);                              00954000010020003
                                                                            00955000
 % The LAST record of firmware code for SBC is transferred to the buffer00956000
 % with the existing NULL padding at the end of that record.  The SBC    00957000
 % parses out those NULLS in its internal buffer before processing the   00958000
 % data.                                                                  00959000
                                                                          00960000
 % The LAST record of Firmware code for drives is transferred to the     00961000
 % buffer with only valid data (no NULLS are included).  The drives      00962000
 % process the entire buffer as valid data.                              00963000
                                                                          00963200010040006
IF FCODEBYTES >= 393216 THEN                                              0096340001.005.000
 BEGIN                                                                    00963600010040006
  I :=0;                                                                  00964500010040006
  SOFAR := 0;                                                             00965000010040006
  WHILE SOFAR < SYSCAP DO                                                 0096600001.005.000
  BEGIN                                                                   0096700001.005.000
%   REPLACE SCR[0] BY "SOFAR= ", SOFAR FOR * DIGITS;                      0096720001.005.000
%   WRITE(RMT,*//,SCR);                                                   0096740001.005.000
%   DISPLAY (SCR);                                                        0096760001.005.001
    READ(CODE,CODEMRS,CODEREC);     % CODEMRS = 8192                      00968000010040006
    PC := CODEREC[0];       % ASSIGN POINTER PC TO CODEREC                00969000010040006
    IF OPTODO = UNASSIGNCTLV THEN           % CODE IS FOR SBC             00970000
      REPLACE IMLBUF2[I,*] BY PC FOR CODEMRS                              00971000010040006
    ELSE                                    % CODE IS FOR DRIVE          00972000
      BEGIN                                                               00973000010040006
        IF (FCODEBYTES - SOFAR) >= CODEMRS THEN  % HAVE FULL DATA RECORD00974000010040006
          REPLACE IMLBUF2[I,*] BY PC FOR CODEMRS                          00975000010040006
        ELSE REPLACE IMLBUF2[I,*] BY PC                                   00976000010040006
                      FOR (FCODEBYTES - SOFAR);          % LAST           00976500010040006
      END;                                               % PARTIAL REC00977000010040006
    I := * + 1;                                                           00977500010040006
    SOFAR := * + CODEMRS;                                                 00978000
  END;                                                                    00979000
  WHILE (FCODEBYTES - SOFAR) > 0 DO                                       0097905001.005.000
  BEGIN                                                                   0097910001.005.000
    I := 0;                                                               0097915001.005.000
    READ(CODE,CODEMRS,CODEREC);                                           0097920001.005.000
    PC := CODEREC[0];                                                     0097925001.005.000
    IF OPTODO = UNASSIGNCTLV THEN                                         0097930001.005.000
      REPLACE IMLBUF3[I,*] BY PC FOR CODEMRS                              0097935001.005.000
    ELSE                                                                  0097940001.005.000
      BEGIN                                                               0097945001.005.000
        IF (FCODEBYTES - SOFAR) >= CODEMRS THEN                           0097950001.005.000
          REPLACE IMLBUF3[I,*] BY PC FOR CODEMRS                          0097955001.005.000
        ELSE                                                              0097960001.005.000
          REPLACE IMLBUF3[I,*] BY PC FOR (FCODEBYTES - SOFAR);            0097965001.005.000
      END;                                                                0097970001.005.000
    I := * + 1;                                                           0097975001.005.000
    SOFAR := * + CODEMRS;                                                 0097980001.005.000
  END;                                                                    0097985001.005.000
END                                                                       00980000010040006
ELSE                                                                      00980050010040006
BEGIN                                                                     00980100010040006
  SOFAR := 0;                                                             00980150010040006
  WHILE SOFAR < FCODEBYTES DO                                             00980200010040006
  BEGIN                                                                   00980250010040006
    READ(CODE,CODEMRS,CODEREC);                                           00980300010040006
    PC := CODEREC[0];                                                     00980350010040006
    IF OPTODO = UNASSIGNCTLV THEN           % CODE IS FOR SBC            00980400010040006
      REPLACE IMLP:IMLP BY PC FOR CODEMRS                                 00980450010040006
    ELSE                                    % CODE IS FOR DRIVE          00980500010040006
    BEGIN                                                                 00980550010040006
    IF (FCODEBYTES - SOFAR) >= CODEMRS THEN  % HAVE FULL DATA RECORD      00980600010040006
```

```
      REPLACE IMLP:IMLP BY PC FOR CODEMRS                                    00980650010040006
    ELSE REPLACE IMLP:IMLP BY PC FOR (FCODEBYTES - SOFAR); % LAST            00980700010040006
    END;                                                   % PARTIAL REC00980750010040006
    SOFAR := * + CODEMRS;                                                   00980800010040006
  END;                                                                      00980850010040006
 END;                                                                       0098090001.002.004
END;                                                                        00980950010040006
END;  % PROCEDURE STOREFWFILEINFO                                           00981000
                                                                            00982000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00983000
%                  Procedure VERIFYFILE                                    %00984000
%                  Procedure VERIFYFILE                                    %0098500001.005.000
                                                                            00986000
PROCEDURE VERIFYFILE;                                                       00987000
BEGIN                                                                       00988000
  INTEGER N, X;                                                             00989000
  LABEL GETCODE;                                                            00990000
                                                                            00991000
% Request name of code file                                                00992000
                                                                            00993000
GETCODE:                                                                    00994000
  CLOSE(CODE);                                                              00995000
  CODE.FILEUSE         := VALUE(IN);                                        00996000
  CODE.KIND            := VALUE(TAPE);                                      00997000
  CODE.DEPENDENTSPECS := TRUE;                                              00998000
  REPLACE TTL[0] BY " ";                                                    00999000
  WHILE TTL[0] = " " DO                                                     01000000
  BEGIN                                                                     01001000
    START                                                                   01002000
      "Enter name of code file or 'Quit'";                                  01003000
    PROMPT;                                                                 01004000
    IF LL > 0 THEN                                                          01005000
    BEGIN                                                                   01006000
      IF PL = "Q" THEN                                                      01007000
        GO GRANDXIT;                                                        01008000
      X := LL;                                                              01009000
      REPLACE L[LL+1] BY ".";                                               01010000
      SCAN PL FOR N:LL UNTIL = ".";                                         01011000
      REPLACE TTL[0] BY PL FOR LL UNTIL = ".", ".";                         01012000
      REPLACE CODE.TITLE BY TTL[0];                                         01013000
      IF CODE.ATTERR THEN                                                   01014000
      BEGIN                                                                 01015000
        BLANK;                                                              01016000
        SHOW("Attribute error setting CODE file title",TRUE);               01017000
        GO GETCODE;                                                         01018000
      END;                                                                  01019000
    END; % LL > 0                                                           01020000
  END; % WHILE                                                              01021000
                                                                            01022000
% Attempt to locate and open code file                                      01023000
                                                                            01024000
% A DISK file must have 'ON <PACKNAME>' in the title declaration            01025000
% (e.g., (UCODE)ABC/123 ON DISK), otherwise it defaults to a TAPE file.     01026000
% If the scan ends with a "." instead of a " ", then a tape type is        01027000
% assummed, because a disk file would require a " " for the "ON PACK"       01028000
% part of the title declaration.                                            01029000
                                                                            01030000
  SCAN PL:PL := TTL[0] FOR N:X UNTIL IN ENDER;                              01031000
                                                                            01032000
  IF PL = "." THEN    % OPEN TAPE                                           01033000
    OPEN (CODE)                      % ALLOW FOR TAPE FILE EQUATE          01034000
  ELSE IF CODE.AVAILABLE = 1 THEN  % DISK FILE                              01035000
    CODEEOF := CODE.LASTRECORD                                              01036000
  ELSE                               % DISK FILE NOT AVAILABLE             01037000
    BEGIN                                                                   01038000
    SHOW ("The disk file name entered is NOT AVAILABLE", TRUE);             01039000
    GO GETCODE;                                                             01040000
    END;                                                                    01041000
                                                                            01042000
  CODEMRS := EIGHTK;                                                        01043000
  IF SIZE(HDR) < CODEMRS THEN                                               01044000
```

```
    BEGIN                                                                   01045000
    RESIZE(HDR,CODEMRS,DISCARD);                                            01046000
    RESIZE(CODEREC,CODEMRS,DISCARD);                                        01047000
    END;                                                                    01048000
                                                                            01049000
    READ(CODE,CODEMRS,CODEREC);                                             01050000
    REPLACE HDR BY CODEREC FOR SIZE(HDR) WITH ASCIITOEBCDIC;                01051000
                                                                            01052000
% See if code file is a valid TARGET code file                             01053000
                                                                            01054000
  IF VALIDATEACODE THEN                                                     01055000
    BEGIN                                                                   01056000
    IF EXPRESSMODE THEN                                                     01056200010020003
      SHOW("File is a valid TARGET code file",FALSE)                        01056400010020003
    ELSE                                                                    01056600010020003
      SHOW("File is a valid TARGET code file",TRUE);                        01057000010020003
    SHOWIT(FALSE);                                                          01058000
    STOREFWFILEINFO;                                                        01059000
    END                                                                     01060000
  ELSE                                                                      01061000
    BEGIN                                                                   01062000
    SHOW("File is -not- a valid TARGET code file",TRUE);                    01063000
    SHOWIT(FALSE);                          % Blank line                   01064000
    GO GETCODE;                                                             01065000
    END;                                                                    01066000
                                                                            01067000
END; % Verify File                                                          01068000
                                                                            01069000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01070000
%      Compare Firmware Levels of Target and Input File                     01071000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01072000
                                                                            01073000
BOOLEAN PROCEDURE COMPFWLEVELS;                                             01074000
BEGIN                                                                       01075000
                                                                            01076000
  INTEGER  FWFILELOC, I;                                                    01077000
  BOOLEAN  FWLEVELSMATCH;                                                   01078000
                                                                            01079000
  I := 0;                                                                   01080000
  % WANT LOWER FOUR BYTES (3,2,1,0) OF 8 BYTE FIELD                        01081000
                                                                            01082000
  FWFILELOC := NUMREPFWLVLSLOC;  % OFFSET BY '4' FOR PROPER LOOPING         01083000
                                 % INCREMENT BELOW.                        01084000
  WHILE (((I:= *+1) LEQ NUMREPFWLVLS) AND (NOT FWLEVELSMATCH)) DO           01085000
    BEGIN                                                                   01086000
    FWLEVELSMATCH := TPRODREVLVL = HDR [(FWFILELOC:= *+ OLDFWLEVELNG)]      01087000
                                   FOR NEWFWLEVELNG;                        01088000
    IF DEBUG THEN                % SHOW ENTIRE 7 OR 8 BYTES OF FILE ENTRY01089000
                                 % IF NULLS ARE PRINTED, OUTPUT IS LOST. 01089100010010001
     IF OPTODO = UNASSIGNUNITV THEN                                         01089200010010001
      BEGIN                                                                 01089400010010001
      IF HDR [(FWFILELOC - 4)] = 48"00" THEN      % BYTE [0] = NULL          01090000
        SHOW ("FILE FW-LOOP = >" C HDR [(FWFILELOC - 3)]                    01091000
                                   FOR (OLDFWLEVELNG - 1) C "<", TRUE)      01092000
      ELSE     % BYTE [0] NEQ NULL. FORMAT REQUIRED FOR SCSI DRIVES         01093000
        SHOW ("FILE FW-LOOP = >" C HDR [(FWFILELOC - 4)]                    01094000
                                   FOR OLDFWLEVELNG  C  "<", TRUE);         01095000
      END                                                                   01095200010010001
     ELSE    % FOR SBC                                                      01095400010010001
        SHOW ("FILE FW-LOOP = >" C HDR [FWFILELOC] FOR NEWFWLEVELNG         01095600010010001
              C "<", TRUE);                                                 01095800010010001
    END;                                                                    01096000
                                                                            01097000
  IF NOT FWLEVELSMATCH THEN      % NO MATCH FOUND                          01098000
     BEGIN                                                                  01099000
     SHOW ("<< Firmware level of target is NOT in input file. >>",TRUE);    01100000
     SHOW ("<< Download will NOT take place. >>", TRUE);                    01101000
     COMPFWLEVELS := FALSE;                                                 01102000
     END                                                                    01103000
  ELSE                                                                      01104000
     BEGIN                                                                  01105000
```

```
      IF OPTODO = UNASSIGNUNITV THEN      % Drives only. Get full 8 bytes
        BEGIN
        REPLACE SAVEFWLVL [0] BY HDR [(FWFILELOC - 4)] FOR OLDFWLEVELNG;
        IF DEBUG THEN
         SHOW ("SAVEFWLVL (FORMAT(IVR) NEEDED?-BYTE[0] NEQ 48'00') = >"
                SAVEFWLVL FOR OLDFWLEVELNG  C  "<", TRUE);
        END;
      COMPFWLEVELS := TRUE;
      END;

 END;  % PROCEDURE COMPFWLEVELS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Compare SERVO Firmware Levels of Target and Input File
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

BOOLEAN PROCEDURE COMPSERVOFWLVLS;
BEGIN

  INTEGER  FWFILELOC, I;
  BOOLEAN  FWLEVELSMATCH;

  I := 0;
  % WANT LOWER FOUR BYTES (3,2,1,0) OF 8 BYTE FIELD

  FWFILELOC := NUMREPFWLVLSLOC;  % OFFSET BY '4' FOR PROPER LOOPING
                                 % INCREMENT BELOW.
  WHILE (((I:= *+1) LEQ NUMREPFWLVLS) AND (NOT FWLEVELSMATCH)) DO
    BEGIN
    FWLEVELSMATCH := TSERVOREVLVL = HDR [(FWFILELOC:= *+ OLDFWLEVELNG)]
                                 FOR NEWFWLEVELNG;
%   SHOW("SERVO FIRMWARE =>" C
%       TSERVOREVLVL FOR NEWFWLEVELNG, TRUE);
    IF DEBUG THEN                % SHOW ENTIRE 7 OR 8 BYTES OF FILE ENTRY
                                 % IF NULLS ARE PRINTED, OUTPUT IS LOST.
     IF OPTODO = UNASSIGNUNITV THEN
      BEGIN
      IF HDR [(FWFILELOC - 4)] = 48"00" THEN      % BYTE [0] = NULL
        SHOW ("FILE FW-LOOP = >" C HDR [(FWFILELOC - 3)]
                                 FOR (OLDFWLEVELNG - 1) C "<", TRUE)
      ELSE     % BYTE [0] NEQ NULL. FORMAT REQUIRED FOR SCSI DRIVES
        SHOW ("FILE FW-LOOP1  = >" C HDR [(FWFILELOC - 4)]
                                 FOR OLDFWLEVELNG  C  "<", TRUE);
      END
     ELSE    % FOR SBC
        SHOW ("FILE FW-LOOP2  = >" C HDR [FWFILELOC] FOR NEWFWLEVELNG
               C "<", TRUE);
    END;

     COMPSERVOFWLVLS := TRUE;

END;  % PROCEDURE COMPFWLEVELS
$$ PAGE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                Release the target
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

BOOLEAN PROCEDURE RELEASETARGET (OPTODO);
VALUE   OPTODO;
INTEGER OPTODO;
BEGIN

  INTEGER  N, RSLT;

% Release the target

  RSLT := USERMAINTREQUEST(CTLUNIT,OPTODO,0,0,0,
                           SHORTBUF,MRD,HDPRESULT);
  IF RSLT > 0 THEN
    BEGIN
    SHOW("Unable to release TARGET "    C
```

```
         CTLUNIT FOR * DIGITS            C
         ".  Error "                     C
         RSLT FOR * DIGITS,TRUE);
    RELEASETARGET := FALSE;
    END
  ELSE
    BEGIN
    RELEASETARGET := TRUE;
    IF OPTODO = UNASSIGNUNITV THEN
      BEGIN
      SDMASSIGNED := FALSE;
      SHOWRSLT (RSLT, UNASSIGNUNITV);
      END
    ELSE
      BEGIN
      CTLASSIGNED := FALSE;
      SHOWRSLT (RSLT, UNASSIGNCTLV);
      END;
    END;

END;  % BOOLEAN PROCEDURE RELEASETARGET
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                Load SBC Firmware Procedure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PROCEDURE LOADCTLFW;
BEGIN
  INTEGER N,RSLT,W,OFFSET,I;
  LABEL NEXTCTL;
  ARRAY T[0:2];

% System call to determine machine type
%  If we are not running on an IOM type system, terminate.

  SYSTEMTYPE := TIME(24);
  REPLACE T BY SYSTEMTYPE FOR 6 ;
  IF T ISNT "A11" FOR 3 AND
     T ISNT "A14" FOR 3 AND
     T ISNT "A12" FOR 3 AND
     T ISNT "A16" FOR 3 AND
     T ISNT "A18" FOR 3 AND
     T ISNT "A19" FOR 3 AND
     T ISNT "A28" FOR 3 AND
     T ISNT "NX46" FOR 4 AND
     T ISNT "NX48" FOR 4 AND
     T ISNT "NX58" FOR 4 THEN
  BEGIN
    SHOW("SYSTEMTYPE = " C SYSTEMTYPE FOR 6, TRUE);
    SHOW("Running system is not an IOM system. Program is ENDING",TRUE);
    GO GRANDXIT;
  END;
  SHOW("SYSTEMTYPE = " C SYSTEMTYPE FOR 6, TRUE);
  START
    "Caution: download FW to only one CTL in a string at a time";
  SHOWIT(TRUE);
  START  "          CTL must be URed to all visible hosts";
  SHOWIT(TRUE);
  START  "          Other CTL must be online";
  SHOWIT(TRUE);

  %****** MAIN TARGET (CTL) LOOP ******%

  OPTODO := UNASSIGNCTLV;     % CTL PARAMETER TO UMR

  WHILE TRUE DO
  BEGIN
  %Request CTL number

  CTLUNIT := 0;
  WHILE CTLUNIT = 0 DO
    BEGIN
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% IF TPRODID EQL "DVC1000-1X1" FOR 11 THEN
    BEGIN
      SHOW("                                                ",TRUE);
      SHOW(">>>>>>>>>>>>>>>>>>ATTENTION<<<<<<<<<<<<<<<<<<<<",TRUE);
    IF NOT ATODT THEN      % AT A TERMINAL
      BEGIN
        SHOW("Download of SBC requires UR of all CTLs ",TRUE);
      END
    ELSE                   % AT AN ODT
      BEGIN
        SHOW("Download of SBC requires UR of all CTLs ",TRUE);
      END;
      SHOW(">>>>>>>>>>>>>>>>>>ATTENTION<<<<<<<<<<<<<<<<<<<<<",TRUE);
      SHOW("                                                ",TRUE);
    END;
%   START "Did you UR all the CTLs & PKs? Enter Yes or No";
%   PROMPT;
%   IF PL NEQ "Y" THEN
%     BEGIN
%       SHOW ("Download will not take place for target " C
%               CTLUNIT FOR * DIGITS, TRUE);
%       IF NOT RELEASETARGET (OPTODO) THEN
%          GO GRANDXIT
%       ELSE
%         GO NEXTCTL;
%     END;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    START
      "Enter number of URed CTL or 'End' to change selection";
    PROMPT;
    IF PL = "E" THEN
      GO MAINLOOP;
    SCAN PL FOR N:LL WHILE >= "0";
    IF LL-N > 0 THEN
      CTLUNIT := INTEGER(PL,LL-N);
    END;

% Validate CTL number as belonging to an SBC

  CTLASSIGNED := FALSE;
  RSLT := USERMAINTREQUEST(CTLUNIT,ASSIGNCTLV,0,0,0,
                           SHORTBUF,MRD,HDPRESULT);
  SHOWRSLT(RSLT,ASSIGNCTLV);
  IF RSLT > 0 THEN                     % Error
  BEGIN
    SHOW("Assign of CTL denied.  Are you sure you ",TRUE);
    SHOW("entered a URed CTL number and not a PORT or PK?",TRUE);
    GO NEXTCTL;
  END
  ELSE
    CTLASSIGNED := TRUE;

% Get attributes of Target

  RSLT := USERMAINTREQUEST(CTLUNIT,ATTRIBUTESV,254,0,0,
                           SHORTBUF,MRD,HDPRESULT);
  IF RSLT > 0 THEN                     % Problem
  BEGIN
    SHOW("MCP interface error " C
         RSLT FOR * DIGITS      C
         " to get target"       C
         " attributes for CTL " C
         CTLUNIT FOR * DIGITS,TRUE);
    SHOWRSLT(RSLT,ATTRIBUTESV);
    SHOW("Check CTL for problem",TRUE);
    IF NOT RELEASETARGET (OPTODO) THEN
      GO GRANDXIT
    ELSE
      GO NEXTCTL;
  END;
```

D22

```
                                                                        01241000
  SHOWATTRIBUTES;                                                       01242000
                                                                        01243000
% Get code file & match header rec. info against Target info.           01244000
                                                                        01245000
  IF TVENDORID NEQ VENDORIDEXP FOR VENDORIDLNG THEN                     01246000
    BEGIN                                                               01247000
    SHOW ("Vendor ID of target is not UNISYS.", TRUE);                  01248000
    SHOW ("<< Download will NOT take place. >>", TRUE);                 01249000
    IF NOT RELEASETARGET (OPTODO) THEN                                  01250000
      GO GRANDXIT                                                       01251000
    ELSE                                                                01252000
      GO NEXTCTL;                                                       01253000
    END;                                                                01254000
  IF NOT COMPPRODIDS THEN   % COMPARE PRODUCT IDs OF FILE VS. TARGET    0125600001001000 1
    IF NOT RELEASETARGET (OPTODO) THEN                                  01257000
      GO GRANDXIT                                                       01258000
    ELSE                                                                01259000
      GO NEXTCTL;                                                       01260000
                                                                        01261000
  IF NOT COMPFWLEVELS THEN  % COMPARE FIRMWARE LEVELS OF FILE VS. TARGET01262000
    IF NOT RELEASETARGET (OPTODO) THEN                                  01263000
      GO GRANDXIT                                                       01264000
    ELSE                                                                01265000
      GO NEXTCTL;                                                       01266000
                                                                        01267000
  IF TPRODREVLVL = FNEWFWLEVEL FOR NEWFWLEVELNG THEN                    01268000
    BEGIN                                                               01269000
    SHOW ("Firmware levels of Target and File are the same.",TRUE);     01270000
    START "Do you still want to download the firmware? Enter YES or NO";01271000
    PROMPT;                                                             01272000
    IF PL NEQ "Y" THEN                                                  01273000
      BEGIN                                                             01274000
      SHOW ("Download will not take place for target " C                 01275000
            CTLUNIT FOR * DIGITS, TRUE);                                01276000
      IF NOT RELEASETARGET (OPTODO) THEN                                01277000
        GO GRANDXIT                                                     01278000
      ELSE                                                              01279000
       GO NEXTCTL;                                                      01280000
      END;                                                              01281000
    END;                                                                01282000
                                                                        01283000
% Request function                                                      01284000
                                                                        01285000
  SHOW("Starting to download code to CTL " C                            01286000
       CTLUNIT FOR * DIGITS,TRUE);                                      01287000
  IF SYSCAP <= FCODEBYTES THEN                                          0128705001004000 6
    BEGIN                                                               0128710001004000 6
      NUMBROFIOS := FCODEBYTES DIV EIGHTK;                              0128715001004000 6
      SIZEOFLSTIO := FCODEBYTES MOD EIGHTK;                             0128720001004000 6
      OFFSET := 0;                                                      0128725001004000 6
      I := 0;                                                           0128730001004000 6
      DO                                                                0128735001004000 6
        BEGIN                                                           0128740001004000 6
          RSLT := USERMAINTREQUEST(CTLUNIT,DOWNLOADMODE6,8192,OFFSET,0, 0128745001004000 6
                        IMLBUF2[I,*],MRD,HDPRESULT);                    0128750001004000 6
          IF RSLT > 0 THEN                                              0128755001004000 6
            BEGIN                                                       0128760001004000 6
              SHOWRSLT(RSLT,LOADSLAVEIMLV);                             0128765001004000 6
              SHOWMRDBITS;                                              0128770001004000 6
              SHOWHDPRESULT;                                            0128775001004000 6
                                                                        0128780001004000 6
              SHOW("<< Microcode NOT loaded!! >>",TRUE);                0128785001004000 6
              IF NOT RELEASETARGET (OPTODO) THEN                        0128790001004000 6
                GO GRANDXIT;                                            0128795001004000 6
            END;                                                        0128800001004000 6
          REPLACE SCR[0] BY "I= ", I FOR *DIGITS;                       0128801001004000 6
          DISPLAY (SCR);                                                0128802001004000 6
          REPLACE SCR2[0] BY "NUMBROFIOS= ", NUMBROFIOS FOR * DIGITS;   0128803001004000 6
          DISPLAY (SCR2);                                               0128804001004000 6
        NUMBROFIOS := * - 1;                                            0128805001004000 6
```

```
          OFFSET := * + EIGHTK;                                          01288100010040006
          I := * + 1;                                                    01288150010040006
        END                                                              01288300010040006
      UNTIL NUMBROFIOS = 1;                                              01288350010040006
      RSLT := USERMAINTREQUEST(CTLUNIT,DOWNLOADMODE7,8192,OFFSET,0,      01288400010040006
                    IMLBUF2[I,*],MRD,HDPRESULT);                         01288450010040006
    END                                                                  01288500010040006
  ELSE                                                                   01288550010040006
  RSLT := USERMAINTREQUEST(CTLUNIT,LOADSLAVEIMLV,FCODEBYTES,0,0,         01288600010040006
                           IMLBUF,MRD,HDPRESULT);                        01288650010040006
  IF RSLT > 0 THEN                                                       01290000
  BEGIN                                                                  01291000
    SHOWRSLT(RSLT,LOADSLAVEIMLV);                                        01292000
    SHOWMRDBITS;;                                                        01293000
    SHOWHDPRESULT;                                                       01294000
                                                                         01295000
    SHOW("<< Microcode NOT loaded!! >>",TRUE);                           01296000
    IF NOT RELEASETARGET (OPTODO) THEN                                   01297000
      GO GRANDXIT                                                        01298000
    ELSE                                                                 01299000
      GO NEXTCTL;                                                        01300000
  END;                                                                   01301000
                                                                         01302000
  SHOW("Download complete.  Waiting to read attributes of CTL " C        01303000
       CTLUNIT FOR * DIGITS,TRUE);                                       01304000
  SHOW("Do not power off or alter CTL " C                                01305000
       CTLUNIT FOR * DIGITS,TRUE);                                       01306000
                                                                         01307000
% The SBC has now turned off its SCSI interface.                        01308000
% It won't come alive until after it has done the power up              01309000
%  confidence tests (approx. 30 seconds).                               01310000
% Initially wait 30 seconds and then cycle ATTRIBUTES calls            01311000
%  every 10 seconds until it comes back to life or drops dead.          01312000
                                                                         01313000
  WHEN(30);                                                              01314000
  SHOW("00:30 Starting to read attributes of CTL " C                     01315000
       CTLUNIT FOR * DIGITS,TRUE);                                       01316000
  TIMER := 0;                                                            01317000
  REPLACE SHORTBUF [0] BY NUL FOR SIZE (SHORTBUF);                       01318000
                                                                         01318050010040006
  RSLT := USERMAINTREQUEST(CTLUNIT,TESTUNITREADY,0,0,0,                  01318100010040006
                           SHORTBUF,MRD,HDPRESULT);                      01318150010040006
                                                                         01318200010040006
          IF RSLT > 0 THEN                                               01318250010040006
            BEGIN                                                        01318300010040006
              SHOWRSLT(RSLT,LOADSLAVEIMLV);                              01318350010040006
              SHOWMRDBITS;                                               01318400010040006
              SHOWHDPRESULT;                                             01318450010040006
                                                                         01318500010040006
              SHOW("<< Test Unit Ready Failed!! >>",TRUE);               01318550010040006
              IF NOT RELEASETARGET (OPTODO) THEN                         01318600010040006
                GO GRANDXIT;                                             01318650010040006
            END;                                                         01318700010040006
DISPLAY(" Test Unit Ready - OK ");                                       01318750010040006
                                                                         01319000
  DO BEGIN                                                               01320000
    WHEN(10);                                                            01321000
    RSLT := USERMAINTREQUEST(CTLUNIT,ATTRIBUTESV,254,0,0,                01322000
                             SHORTBUF,MRD,HDPRESULT);                    01323000
    N := (TIMER:=*+1)*10 + 30;          % Seconds                       01324000
    SHOWRSLT(RSLT,-N);                                                   01325000
                                                                         01326000
  END UNTIL N >=  1*60                  % Drop dead time                01327000
         OR RSLT = 0;                   % Success                       01328000
                                                                         01329000
  IF RSLT > 0 THEN                      % Timeout                       01330000
  BEGIN                                                                  01331000
    SHOW("MCP interface error " C                                        01332000
         RSLT FOR * DIGITS      C                                        01333000
         " after FW download"   C                                        01334000
         " timeout on CTL "     C                                        01335000
```

```
        CTLUNIT FOR * DIGITS,TRUE);                                          01336000
    SHOWRSLT(RSLT,ATTRIBUTESV);                                              01337000
    SHOW("Check CTL for malfunction",TRUE);                                  01338000
    GO GRANDXIT;                                                             01339000
  END;                                                                       01340000
                                                                             01341000
  SHOW("CTL "                    C                                         01342000
       CTLUNIT FOR * DIGITS      C                                         01343000
       " completed FW download",TRUE);                                       01344000
                                                                             01348000
% Format attributes                                                         01349000
                                                                             01350000
  SHOWATTRIBUTES;                                                          01351000
                                                                             01351100010010001
  SHOW ("                                                  ", TRUE);       01351200010010001
  SHOW ("If BOOT code was loaded instead of OPERATIONAL code", TRUE);       01351300010010001
  SHOW ("The firmware level of the SBC may not have changed ", TRUE);       01351400010010001
  SHOW ("                                                  ", TRUE);       01351500010010001
                                                                             01352000
% Release the CTL                                                           01353000
                                                                             01354000
  IF NOT RELEASETARGET (OPTODO) THEN                                       01355000
    GO GRANDXIT;                                                           01356000
                                                                             01357000
  SHOW("Okay to UR- CTL "        C                                         01357200010020002
       CTLUNIT FOR * DIGITS      C                                         01357400010020002
       " and load companion CTL",TRUE);                                      01357600010020002
                                                                             01357800010020002
NEXTCTL:                                                                   01358000
  END;  % WHILE TRUE DO                                                    01359000
                                                                             01360000
END; % Load CTL Firmware                                                   01361000
                                                                             01362000
                                                                             01363000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01364000
%               Load SCSI Drive Firmware Procedure                         0136500001.005.001
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01366000
                                                                             01367000
PROCEDURE LOADEVFW;                                                        0136800001.005.001
BEGIN                                                                      01369000
  INTEGER N,RSLT,W,OFFSET,I,J;                                             0137000001.005.001
  LABEL NEXTDRIVE;                                                         01371000
  ARRAY T[0:2];                                                            01372000010030005
% System call to determine machine type                                    01373000
%  If we are not running on an IOM type system, terminate.                 01374000
                                                                             01375000
  SYSTEMTYPE := TIME(24);                                                  01376000010030005
% REPLACE T BY SYSTEMTYPE FOR 3 ;                                          01376500010030005
% IF T ISNT "A11" FOR 3 AND                                                01377000010030005
%    T ISNT "A14" FOR 3 AND                                                01377500010030005
%    T ISNT "A16" FOR 3 AND                                                01378000010030005
%    T ISNT "A17" FOR 3 AND                                                01379000010030005
%    T ISNT "A18" FOR 3 AND                                                01379500010030005
%    T ISNT "A19" FOR 3 AND                                                01380000010030005
%    T ISNT "A28" FOR 3 THEN                                               01380500010030005
% BEGIN                                                                    01381000010030005
%   SHOW("SYSTEMTYPE = " C SYSTEMTYPE FOR 6, TRUE);                        01381500010030005
%   SHOW("Running system does not support this drives."  C                 01382000010030005
%        " Program is ENDING.",TRUE);                                      01383000010030005
%   GO GRANDXIT;                                                           01384000010030005
% END;                                                                     01385000010030005
  SHOW("SYSTEMTYPE = " C SYSTEMTYPE FOR 6,TRUE);                           01386000010030005
  IF NOT EXPRESSMODE THEN                                                  01386200010020003
    BEGIN                                                                  01386400010020003
    START  "Caution: Load only one drive in a string at a time";           01387000010020003
    SHOWIT(TRUE);                                                          01388000010020003
    START  "         drive must be URed to all visible hosts";             01389000010020003
    SHOWIT(TRUE);                                                          01390000010020003
    START  "         If using Fibre Channel UR the opposite Port";         0139020001.005.000
    SHOWIT(TRUE);                                                          0139040001.005.000
    END;                                                                   01390500010020003
```

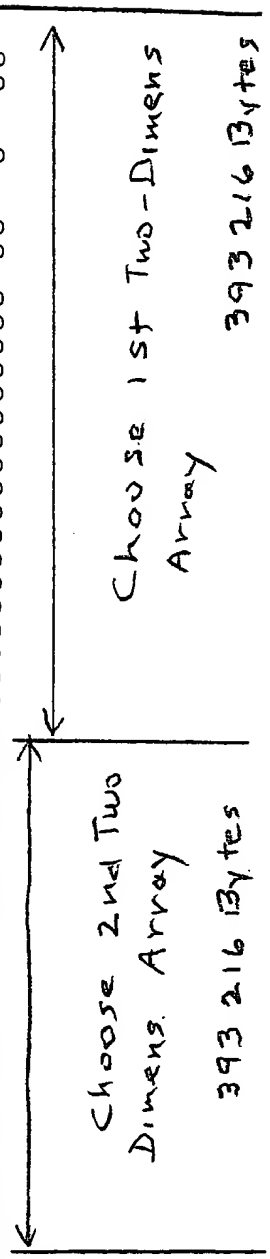
```
%****** MAIN TARGET (DRIVE) LOOP ******%

OPTODO := UNASSIGNUNITV;     % DRIVE PARAMETER TO UMR

WHILE TRUE DO
BEGIN

%Request drive number

CTLUNIT := 0;
WHILE CTLUNIT = 0 DO
  BEGIN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   BEGIN
%     SHOW("                                                   ",TRUE);
%     SHOW(">>>>>>>>>>>>>>>>>>>ATTENTION<<<<<<<<<<<<<<<<<<<<<",TRUE);
%   IF NOT ATODT THEN      % AT A TERMINAL
%     BEGIN
%       SHOW("All PKs associated with a DRC physical    ",TRUE);
%       SHOW("disk must be CLOSED and UR'ed.            ",TRUE);
%     END
%   ELSE                   % AT AN ODT
%     BEGIN
%       SHOW("disk must be CLOSED and UR'ed.            ",TRUE);
%       SHOW("All PKs associated with a DRC physical    ",TRUE);
%     END;
%     SHOW(">>>>>>>>>>>>>>>>>>>ATTENTION<<<<<<<<<<<<<<<<<<<<<",TRUE);
%     SHOW("                                                   ",TRUE);
%   END;
%   START "Did you UR all the Packs associated? Enter Yes or No";
%   PROMPT;
%   IF PL NEQ "Y" THEN
%     BEGIN
%       SHOW ("Download will not take place for target " C
%              CTLUNIT FOR * DIGITS, TRUE);
%       IF NOT RELEASETARGET (OPTODO) THEN
%          GO GRANDXIT;
%       ELSE
%         GO NEXTDRIVE;
%     END;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    START
      "Enter number of URed drive or 'End' to change selection.";
    PROMPT;
    IF PL = "E" THEN
      GO MAINLOOP;
    SCAN PL FOR N:LL WHILE >= "0";
    IF LL-N > 0 THEN
      CTLUNIT := INTEGER(PL,LL-N);
    END;

% Validate SCSI disk drive number

  SDMASSIGNED := FALSE;
  RSLT := USERMAINREQUEST(CTLUNIT,ASSIGNUNITV,0,0,0,
                          SHORTBUF,MRD,HDPRESULT);
  SHOWRSLT(RSLT,ASSIGNUNITV);
  IF RSLT > 0 THEN                        % Error
  BEGIN
    SHOW("Assign of the drive denied.  Are you sure you ",TRUE);
    SHOW("entered a URed drive number and not a PORT or CTL?",TRUE);
    GO NEXTDRIVE;
  END
  ELSE
    SDMASSIGNED := TRUE;

% Get attributes of Target

  RSLT := USERMAINREQUEST(CTLUNIT,ATTRIBUTESV,254,0,0,
                          SHORTBUF,MRD,HDPRESULT);
```

```
  IF RSLT > 0 THEN                        % Problem                        01433000
  BEGIN                                                                   01434000
    SHOW("MCP interface error "          C                                01435000
         RSLT FOR * DIGITS               C                                01436000
         " to get target"                C                                01437000
         " attributes for SCSI drive " C                                  01438000
         CTLUNIT FOR * DIGITS,TRUE);                                      01439000
    SHOWRSLT(RSLT,ATTRIBUTESV);                                           01440000
    SHOW("Check TARGET for problem",TRUE);                                01441000
    IF NOT RELEASETARGET (OPTODO) THEN                                    01442000
      GO GRANDXIT                                                         01443000
    ELSE                                                                  01444000
      GO NEXTDRIVE;                                                       01445000
  END;                                                                    01446000
                                                                          01447000
  SHOWATTRIBUTES;                                                         01448000
                                                                          01449000
  IF EXPRESSMODE THEN         % Save for display at end of download       01449200010020003
    REPLACE OLDFWLEVEL [0] BY TPRODREVLVL FOR NEWFWLEVELNG;               01449400010020003
                                                                          01449600010020003
% Get code file & match header rec. info against Target info.             01450000
                                                                          01451000
  IF TVENDORID NEQ VENDORIDEXP FOR VENDORIDLNG THEN                       01452000
    BEGIN                                                                 01453000
    SHOW ("Vendor ID of target is not UNISYS.", TRUE);                    01454000
    SHOW ("<< Download will NOT take place. >>", TRUE);                   01455000
    IF NOT RELEASETARGET (OPTODO) THEN                                    01456000
      GO GRANDXIT                                                         01457000
    ELSE                                                                  01458000
      GO NEXTDRIVE;                                                       01459000
    END;                                                                  01460000
  IF NOT COMPPRODIDS THEN   % COMPARE PRODUCT IDs OF FILE vs. TARGET      01462000010010001
    IF NOT RELEASETARGET (OPTODO) THEN                                    01463000
      GO GRANDXIT                                                         01464000
    ELSE                                                                  01465000
      GO NEXTDRIVE;                                                       01466000
                                                                          01467000
  IF TPRODREVLVL = FNEWFWLEVEL  FOR NEWFWLEVELNG THEN                     0147400001.005.000
    BEGIN                                                                 01475000
    SHOW ("Firmware levels of Target and File are the same.",TRUE);       01476000
    START "Do you still want to download the firmware? Enter YES or NO";  01477000
    PROMPT;                                                               01478000
    IF PL NEQ "Y" THEN                                                    01479000
      BEGIN                                                               01480000
      SHOW ("Download will not take place for target " C                  01481000
            CTLUNIT FOR * DIGITS, TRUE);                                  01482000
      IF NOT RELEASETARGET (OPTODO) THEN                                  01483000
        GO GRANDXIT                                                       01484000
      ELSE                                                                01485000
       GO NEXTDRIVE;                                                      01486000
      END                                                                 01487000
    ELSE REPLACE SAVEFWLVL [0] BY 48"00" FOR OLDFWLEVELNG;   % NO FORMAT  01487500010010001
    END                                                                   01488000
  ELSE                                                                    01489000
  IF NOT COMPFWLEVELS THEN  % COMPARE FW LEVELS OF FILE VS. TARGET        0148920001.005.000
    IF NOT RELEASETARGET (OPTODO) THEN                                    01489400010010001
      GO GRANDXIT                                                         01489600010010001
    ELSE                                                                  01489800010010001
      GO NEXTDRIVE;                                                       01490000010010001
                                                                          01490200010010001
  % If drive needs to be formatted after firmware download, get           01491000010010001
  % permission before downloading the code.                              01492000010010001
  % Each FW level area in file = 8 bytes. If byte 0 neq 48"00", drive     01493000010010001
  % must be formatted after code is loaded.                              01494000010010001
  IF SAVEFWLVL [0] NEQ 48"00" THEN   % Drive needs formatting            01495000010010001
    IF NOT OKTOFORMAT THEN                                                01496000010010001
      IF NOT RELEASETARGET (OPTODO) THEN                                  01497000010010001
        GO GRANDXIT                                                       01498000
       ELSE                                                               01499000
        GO NEXTDRIVE;                                                     01500000
                                                                          01502000
```

```
% Get attributes of Target                                                0150201001.005.001
                                                                         0150202001.005.001
  RSLT := USERMAINTREQUEST(CTLUNIT,INQUIRY,23,192,0,                     0150203001.005.000
                           SHORTBUF,MRD,HDPRESULT);                      0150204001.005.001
  IF RSLT > 0 THEN                        % Problem                       0150205001.005.001
  BEGIN                                                                  0150206001.005.001
    SHOW("MCP interface error "        C                                 0150207001.005.001
         RSLT FOR * DIGITS             C                                 0150208001.005.001
         " to get target"              C                                 0150209001.005.001
         " attributes for SCSI drive " C                                 0150210001.005.001
         CTLUNIT FOR * DIGITS,TRUE);                                     0150211001.005.001
    SHOWRSLT(RSLT,ATTRIBUTESV);                                          0150212001.005.001
    SHOW("Check TARGET for problem",TRUE);                               0150213001.005.001
    IF NOT RELEASETARGET (OPTODO) THEN                                   0150214001.005.001
      GO GRANDXIT                                                        0150215001.005.001
    ELSE                                                                 0150216001.005.001
      GO NEXTDRIVE;                                                      0150217001.005.001
  END;                                                                   0150218001.005.001
                                                                         0150219001.005.001
  SHOWINQUIRYDATA;                                                       0150220001.005.000
                                                                         0150221001.005.001
  IF EXPRESSMODE THEN        % Save for display at end of download       0150222001.005.001
    REPLACE OLDFWLEVEL [0] BY TSERVOREVLVL FOR NEWFWLEVELNG;             0150223001.005.000
                                                                         0150224001.005.001
% Get code file & match header rec. info against Target info.            0150225001.005.001
                                                                         0150226001.005.001
                                                                         0150241001.005.001
  IF TSERVOREVLVL  = FNEWFWLEVEL FOR NEWFWLEVELNG THEN                   0150242001.005.000
    BEGIN                                                                0150243001.005.001
    SHOW ("Servo FW levels of Target and File are the same.",TRUE);      0150244001.005.000
    START "Do you still want to download the firmware? Enter YES or NO";0150245001.005.001
    PROMPT;                                                              0150246001.005.001
    IF PL NEQ "Y" THEN                                                   0150247001.005.001
      BEGIN                                                              0150248001.005.001
      SHOW ("Download will not take place for target " C                 0150249001.005.001
            CTLUNIT FOR * DIGITS, TRUE);                                 0150250001.005.001
      IF NOT RELEASETARGET (OPTODO) THEN                                 0150251001.005.001
        GO GRANDXIT                                                      0150252001.005.001
      ELSE                                                               0150253001.005.001
       GO NEXTDRIVE;                                                     0150254001.005.001
      END                                                                0150255001.005.001
    ELSE REPLACE SAVEFWLVL [0] BY 48"00" FOR OLDFWLEVELNG;   % NO FORMAT0150256001.005.001
    END                                                                  0150257001.005.001
  ELSE                                                                   0150258001.005.001
  IF NOT COMPSERVOFWLVLS  THEN  % COMPARE FW LEVELS OF FILE VS. TARGET   0150259001.005.000
    IF NOT RELEASETARGET (OPTODO) THEN                                   0150260001.005.001
      GO GRANDXIT                                                        0150261001.005.001
    ELSE                                                                 0150262001.005.001
      GO NEXTDRIVE;                                                      0150263001.005.001
                                                                         0150264001.005.001
  % If drive needs to be formatted after firmware download, get          0150265001.005.001
  % permission before downloading the code.                              0150266001.005.001
  % Each FW level area in file = 8 bytes. If byte 0 neq 48"00", drive    0150267001.005.001
  % must be formatted after code is loaded.                             0150268001.005.001
  IF SAVEFWLVL [0] NEQ 48"00" THEN   % Drive needs formatting            0150269001.005.001
    IF NOT OKTOFORMAT THEN                                               0150270001.005.001
      IF NOT RELEASETARGET (OPTODO) THEN                                 0150271001.005.001
        GO GRANDXIT                                                      0150272001.005.001
       ELSE                                                              0150273001.005.001
        GO NEXTDRIVE;                                                    0150274001.005.001
                                                                         0150275001.005.001
% Request function                                                       01503000
                                                                         01504000
  IF NOT EXPRESSMODE THEN                                                01504500010020003
    SHOW("Starting to download code to drive " C                         01505000010020003
         CTLUNIT FOR * DIGITS,TRUE);                                     01506000010020003
  IF FCODEBYTES > SYSCAP THEN                                            0150610001.005.000
    BEGIN                                                                01506200010040006
      NUMBROFIOS := FCODEBYTES DIV EIGHTK;                               01506300010040006
%     NUMBROFIOS := * + 1;                                               0150632001.005.000
%     SHOW(" NUMBROFIOS = " C NUMBROFIOS FOR * DIGITS,TRUE);             0150635001.005.000
```

DECISION ON SYSTEM BYTES

```
      SIZEOFLSTIO := FCODEBYTES MOD EIGHTK;                                  01506400010040006
%     SHOW(" SIZEOFLSTIO =  " C SIZEOFLSTIO FOR * DIGITS,TRUE);              0150640201.005.000
      FIRSTTIME := TRUE;                                                     0150640401.005.000
      OFFSET := 0;                                                           01506410010040006
      I := 0;                                                                01506415010040006
      J := 0;                                                                0150641601.005.000
      DO                                                                     01506420010040006
        BEGIN                                                                01506430010040006
         IF J <= 47 THEN                                                     0150643501.005.000
         BEGIN                                                               0150643601.005.000
          RSLT := USERMAINTREQUEST(CTLUNIT,DOWNLOADMODE7,8192,OFFSET,0,      01506440010040006
                        IMLBUF2[I,*],MRD,HDPRESULT);                         0150645001.005.000
%         SHOW("1I = " C I FOR * DIGITS,TRUE);                               0150645201.005.000
%         SHOW("1OFFSET = " C OFFSET FOR * DIGITS,TRUE);                     0150645401.005.000
          IF RSLT > 0 THEN                                                   0150645601.005.000
            BEGIN                                                            0150645801.005.000
              SHOWRSLT(RSLT,LOADSLAVEIMLV);                                  0150646001.005.000
              SHOWMRDBITS;                                                   0150646201.005.000
              SHOWHDPRESULT;                                                 0150646401.005.000
                                                                             0150646601.005.000
              SHOW("<< Microcode(1) NOT loaded!! >>",TRUE);                  0150646801.005.000
              IF NOT RELEASETARGET (OPTODO) THEN                             0150647001.005.000
                GO GRANDXIT;                                                 0150647201.005.000
            END;                                                             0150647401.005.000
          END                                                                0150647601.005.000
         ELSE                                                                0150647801.005.000
          BEGIN                                                              0150648001.005.000
          IF FIRSTTIME THEN                                                  0150648201.005.000
            I := 0;                                                          0150648401.005.000
          FIRSTTIME := FALSE;                                                0150648601.005.000
          RSLT := USERMAINTREQUEST(CTLUNIT,DOWNLOADMODE7,8192,OFFSET,0,      0150648801.005.000
                        IMLBUF3[I,*],MRD,HDPRESULT);                         0150649001.005.000
%         SHOW("2I = " C I FOR * DIGITS,TRUE);                               0150649201.005.000
%         SHOW("2OFFSET = " C OFFSET FOR * DIGITS,TRUE);                     0150649401.005.000
%         SHOW("2J = " C J FOR * DIGITS,TRUE);                               0150649601.005.000
          IF RSLT > 0 THEN                                                   0150649801.002.006
            BEGIN                                                            0150650001.002.006
              SHOWRSLT(RSLT,LOADSLAVEIMLV);                                  0150650201.002.006
              SHOWMRDBITS;                                                   0150650401.002.006
              SHOWHDPRESULT;                                                 0150650601.002.006
                                                                             0150650801.002.006
              SHOW("<< Microcode(2) NOT loaded!! >>",TRUE);                  0150651001.005.000
              IF NOT RELEASETARGET (OPTODO) THEN                             0150651201.002.006
                GO GRANDXIT;                                                 0150651401.002.006
            END;                                                             0150651601.002.006
          END;                                                               0150651801.005.000
          NUMBROFIOS := * - 1;                                               0150652001.002.006
          OFFSET := * + EIGHTK;                                              0150652201.002.006
          I := * + 1;                                                        0150652401.002.006
          J := * + 1;                                                        0150652601.005.000
        END                                                                  0150652801.002.006
      UNTIL NUMBROFIOS = 0;                                                  0150653001.005.000
%     SHOW(" NUMBROFIOS = " C NUMBROFIOS FOR * DIGITS,TRUE);                 0150653201.005.000
%     SHOW(" I = " C I FOR * DIGITS,TRUE);                                   0150653401.005.000
%     SHOW(" J = " C J FOR * DIGITS,TRUE);                                   0150653601.005.000
%     SHOW(" OFFSET = " C OFFSET FOR * DIGITS,TRUE);                         0150653801.005.000
      IF J < 47 THEN                                                         0150654001.005.000
        BEGIN                                                                0150654201.005.000
         RSLT := USERMAINTREQUEST(CTLUNIT,DOWNLOADMODE7,8192,OFFSET,0,       0150654401.005.000
                       IMLBUF2[I,*],MRD,HDPRESULT);                          0150655001.005.000
%         SHOW("3I = " C I FOR * DIGITS,TRUE);                               0150655201.005.000
%         SHOW("3OFFSET = " C OFFSET FOR * DIGITS,TRUE);                     0150655401.005.000
          IF RSLT > 0 THEN                                                   0150655601.005.000
            BEGIN                                                            0150655801.005.000
              SHOWRSLT(RSLT,LOADSLAVEIMLV);                                  0150656001.005.000
              SHOWMRDBITS;                                                   0150656201.005.000
              SHOWHDPRESULT;                                                 0150656401.005.000
                                                                             0150656601.005.000
              SHOW("<< Microcode(3) NOT loaded!! >>",TRUE);                  0150656801.005.000
              IF NOT RELEASETARGET (OPTODO) THEN                             0150657001.005.000
                GO GRANDXIT;                                                 0150657201.005.000
```



```
          END;                                                          0150657401.005.000
        END                                                             0150657601.005.000
      ELSE                                                              0150657801.005.000
        BEGIN                                                           0150658001.005.000
         RSLT := USERMAINTREQUEST(CTLUNIT,DOWNLOADMODE7,4224,OFFSET,0,  0150658201.005.000
                  IMLBUF3[I,*],MRD,HDPRESULT);                          0150658401.005.000
%         SHOW("4I = " C I FOR * DIGITS,TRUE);                          0150658601.005.000
%         SHOW("4OFFSET = " C OFFSET FOR * DIGITS,TRUE);                0150658801.005.000
%         SHOW("4J = " C J FOR * DIGITS,TRUE);                          0150659001.005.000
          IF RSLT > 0 THEN                                              0150660001.005.000
            BEGIN                                                       0150662001.005.000
              SHOWRSLT(RSLT,LOADSLAVEIMLV);                             0150664001.005.000
              SHOWMRDBITS;                                              0150666001.005.000
              SHOWHDPRESULT;                                            0150668001.005.000
                                                                        0150670001.005.000
              SHOW("<< Microcode(4) NOT loaded!! >>",TRUE);             0150672001.005.000
              IF NOT RELEASETARGET (OPTODO) THEN                        0150674001.005.000
                GO GRANDXIT;                                            0150676001.005.000
            END;                                                        0150678001.005.000
        END;                                                            0150680001.005.000
    END                                                                 0150682001.002.006
  ELSE                                                                  0150684001.002.006
  RSLT := USERMAINTREQUEST(CTLUNIT,LOADSLAVEIMLV,FCODEBYTES,0,0,        01507000
                           IMLBUF,MRD,HDPRESULT);                       01508000010040006
%         SHOW("5I = " C I FOR * DIGITS,TRUE);                          0150820001.005.000
%         SHOW("5OFFSET = " C OFFSET FOR * DIGITS,TRUE);                0150840001.005.000
  IF RSLT > 0 THEN                                                      01509000
  BEGIN                                                                 01510000
    SHOWRSLT(RSLT,LOADSLAVEIMLV);                                       01511000
    SHOWMRDBITS;;                                                       01512000
    SHOWHDPRESULT;                                                      01513000
                                                                        01514000
    SHOW("<< Microcode(5) NOT loaded!! >>",TRUE);                       0151500001.005.000
    IF NOT RELEASETARGET (OPTODO) THEN                                  01516000
      GO GRANDXIT                                                       01517000
    ELSE                                                                01518000
      GO NEXTDRIVE;                                                     01519000
  END;                                                                  01520000
                                                                        01521000
  IF EXPRESSMODE THEN                                                   01521200010020003
    SHOW("Download complete. Waiting 20 seconds for prom burn to " C    01521400010020003
         CTLUNIT FOR * DIGITS,TRUE)                                     01521600010020003
  ELSE                                                                  01521800010020003
    SHOW("Download complete.  Waiting 60 seconds for prom burn to " C   01522000010020003
         CTLUNIT FOR * DIGITS,TRUE);                                    01523000010020003
  SHOW("Do not power off or alter drive " C                             01524000
       CTLUNIT FOR * DIGITS,TRUE);                                      01525000
                                                                        01526000
% The SCSI disk drive has now turned off its SCSI interface.            01527000
% It won't come alive until after it has done the power up              01528000
% confidence tests (approx. 60 seconds). The second ATTRIBUTES command  01529000
% can then be issued to display the new firmware level.  NOTE: If the   01530000
% second ATTRIBUTES command is issued before the target sequences       01531000
% (powers itself back up) the target will hang and the program will show01532000
% an error.                                                             01533000
                                                                        01534000
  IF NOT EXPRESSMODE THEN                                               01534500010020003
    SHOW("00:10 - Waiting for prom burn to target " C                   01535000010020003
         CTLUNIT FOR * DIGITS,TRUE);                                    01536000010020003
  TIMER := 0;                                                           01537000
                                                                        01538000
  IF NOT EXPRESSMODE THEN                                               01538500010020003
  DO BEGIN                                                              01539000
    WHEN(10);                                                           01540000
    N := (TIMER:=*+1)*10 + 10;          % Seconds                       01541000
    IF NOT EXPRESSMODE THEN                                             01541500010020003
      SHOWRSLT(RSLT,-N);                                                01542000010020003
  END UNTIL N >=  1*60;                 % Drop dead time                01543000
                                                                        01543200010020003
  IF EXPRESSMODE THEN                   % WAIT 20 SECONDS ONLY          01543400010020003
    WHEN (20);                                                          01543600010020003
```

LOAD SINGLE ARRAY

```
  REPLACE SHORTBUF [0] BY NUL FOR SIZE (SHORTBUF);

  RSLT := USERMAINTREQUEST(CTLUNIT,TESTUNITREADY,0,0,0,
                           SHORTBUF,MRD,HDPRESULT);

          IF RSLT > 0 THEN
            BEGIN
              SHOWRSLT(RSLT,LOADSLAVEIMLV);
              SHOWMRDBITS;
              SHOWHDPRESULT;

              SHOW("<< Test Unit Ready Failed!! >>",TRUE);
              IF NOT RELEASETARGET (OPTODO) THEN
                GO GRANDXIT;
            END;
  DISPLAY(" Test Unit Ready - OK ");
  REPLACE SHORTBUF [0] BY NUL FOR SIZE (SHORTBUF);

  RSLT := USERMAINTREQUEST(CTLUNIT,ATTRIBUTESV,254,0,0,
                           SHORTBUF,MRD,HDPRESULT);

  IF RSLT > 0 THEN                          % Timeout
    BEGIN
    SHOW("MCP interface error "    C
         RSLT FOR * DIGITS         C
         " after FW download"      C
         " timeout on drive "      C
         CTLUNIT FOR * DIGITS,TRUE);
    SHOWRSLT(RSLT,ATTRIBUTESV);
    SHOW("Check the drive for malfunction",TRUE);
    IF EXPRESSMODE THEN
      SHOW(">> Run DFAST again NOT using Express Mode <<",TRUE);
    GO GRANDXIT;
    END;

  SHOW("Target " C CTLUNIT FOR * DIGITS C " completed download",TRUE);

  SHOWATTRIBUTES;

% Get attributes of Target

  RSLT := USERMAINTREQUEST(CTLUNIT,INQUIRY,23,192,0,
                           SHORTBUF,MRD,HDPRESULT);
  IF RSLT > 0 THEN                          % Problem
  BEGIN
    SHOW("MCP interface error "          C
         RSLT FOR * DIGITS               C
         " to get target"                C
         " attributes for SCSI drive " C
         CTLUNIT FOR * DIGITS,TRUE);
    SHOWRSLT(RSLT,ATTRIBUTESV);
    SHOW("Check TARGET for problem",TRUE);
    IF NOT RELEASETARGET (OPTODO) THEN
      GO GRANDXIT
    ELSE
      GO NEXTDRIVE;
  END;

  SHOWINQUIRYDATA;

  IF EXPRESSMODE THEN
    SHOW("ProdID-" C TPRODID FOR DRIVEIDLNG C "OldFWlvl-" C
         OLDFWLEVEL [0] FOR NEWFWLEVELNG    C ".NewFWlvl-" C
         TPRODREVLVL    FOR NEWFWLEVELNG,TRUE);

% Release the target drive

  IF NOT RELEASETARGET (OPTODO) THEN
    GO GRANDXIT;
```

```
01543620010040006
01543640010040006
01543660010040006
01543680010040006
01543700010040006
01543720010040006
01543740010040006
01543760010040006
01543780010040006
01543800010040006
01543820010040006
01543840010040006
01543860010040006
01543880010040006
01543900010040006
01544000010040006
01544500010040006
01545000
01546000
01547000
01548000
01549000
01550000
01551000010020003
01552000
01553000
01554000
01555000
01556000
01557000
01558000
01558200010020003
01558400010020003
01559000
01560000010020003
01561000
01563000010020003
01565000
01566000
01571000
0157100201.005.000
0157100401.005.000
0157100601.005.000
0157100801.005.000
0157101001.005.000
0157101201.005.000
0157101401.005.000
0157101601.005.000
0157101801.005.000
0157102001.005.000
0157102201.005.000
0157102401.005.000
0157102601.005.000
0157102801.005.000
0157103001.005.000
0157103201.005.000
0157103401.005.000
0157103601.005.000
0157103801.005.000
0157104001.005.000
0157104201.005.000
01571100010020003
01571200010020003
01571300010020003
01571400010020003
01571500010020003
01572000
01573000
01574000
01575000
01575200010020002
```

```
  SHOW("Okay to UR- target "    C
       CTLUNIT FOR * DIGITS     C
       " and select another drive.",TRUE);

NEXTDRIVE:
  END;  % WHILE TRUE DO

END; % Load SCSI drive Firmware
$$ PAGE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Initialization Block
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

PROCEDURE INITIALIZE;
BEGIN

  INTEGER N;

  ODTMODE      := MYSELF.INITIATOR >= 0;
  ATODT        := MYSELF.INITIATOR  = 0;

  SDMASSIGNED :=
  CTLASSIGNED := FALSE;

  TSTMP[0]  := COMPILETIME(15);           % mmddyy
  TSTMP[1]  := COMPILETIME(09);           % hhmmss
  TSTMP[2]  := TIME(15);                  % mmddyy
  TSTMP[3]  := TIME(09);                  % hhmmss

% See if we are privileged and can make use of the MCP interface.

  N := LINKLIBRARY(MCP);
  IF DEBUG THEN
  BEGIN
    START "MCP linkage result = ",
          N FOR * NUMERIC;
    IF N = 1 THEN
      ADD " (normal)";
    SHOWIT(TRUE);
  END;
  IF N < 0 THEN
  BEGIN
    SHOW("FW download not possible.  Program is ENDING.",TRUE);
    GO GRANDXIT;
  END;

  BLANK;
  START
    MYSELF.NAME,
    " (Version ",
    COMPILETIME(20) FOR 2 DIGITS,
    ".",
    COMPILETIME(21) FOR 3 DIGITS,
    ") compiled ",
    ETSTMP[00] WITH DATENTIME,
    ", run on ",
    ETSTMP[12] WITH DATENTIME;
  SHOWIT(TRUE);
  SHOWIT(FALSE);                          % Blank line

  START
  "DO YOU WANT TO VIEW THE USER DOCUMENTATION FOR DFAST? 'Y' OR 'N'";
  PROMPT;
  IF PL = "Y" THEN
    USERPROCESS
  ELSE
    READFCN;

  START
    "  >> Class C Material.  Proprietary to Unisys Corporation <<";
```

```
  SHOWIT(TRUE);                                                          01642000
  SHOWIT(FALSE);                                                         01643000
                                                                         01644000
$$ SET OMIT                                                              01645000
                                                                         01646000
  START                                                                  01647000
    "  >>**************  A T T E N T I O N  *****************<<  ";       01648000
  SHOWIT(TRUE); SHOWIT(FALSE);                                           01649000
%*IF NOT ATODT THEN              % AT A TERMINAL                        01650000
%*  BEGIN                                                              01651000
    START                                                                01652000
      "  >> This is an U N O F F I C I A L version of the FW  <<   ";     01653000
    SHOWIT(TRUE);  SHOWIT(FALSE);                                        01654000
    START                                                                01655000
      "  >> download program to be used for project development << ";     01656000
    SHOWIT(TRUE);  SHOWIT(FALSE);                                        01657000
    START                                                                01658000
      "  >> ONLY.  Do not pass this file to other organizations. <<";    01659000
    SHOWIT(TRUE);  SHOWIT(FALSE);                                        01660000
%*  END                                                                01661000
% $$ SET OMIT                                                          01662000
  ELSE                           % AT AN A-SERIES CONSOLE               01663000
    BEGIN                                                                01664000
    START                                                                01665000
      "  >> ONLY.  Do not pass this file to other organizations. <<";    01666000
    SHOWIT(TRUE);  SHOWIT(FALSE);                                        01667000
    START                                                                01668000
      "  >> download program to be used for project development << ";     01669000
    SHOWIT(TRUE);  SHOWIT(FALSE);                                        01670000
    START                                                                01671000
      "  >> This is an U N O F F I C I A L version of the FW  <<   ";     01672000
    SHOWIT(TRUE);  SHOWIT(FALSE);                                        01673000
    END;                                                                 01674000
% $$ POP OMIT                                                           01675000
                                                                         01676000
  START                                                                  01677000
    "  >>**************  A T T E N T I O N  *****************<<  ";       01678000
  SHOWIT(TRUE); SHOWIT(FALSE);                                           01679000
                                                                         01680000
$$ POP OMIT                                                              01681000
                                                                         01682000
 END;  % PROCEDURE INITIALIZE                                            01683000
                                                                         01684000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01685000
%                    Select the Option Desired                          %01686000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01687000
                                                                         01688000
 PROCEDURE SELECTOPTION;                                                 01689000
 BEGIN                                                                   01690000
                                                                         01691000
 % Determine our operating mode                                          01692000
                                                                         01693000
  FUNCTION     := 0;                                                     01694000010020003
  EXPRESSMODE := FALSE;                                                  01694200010020003
  DEBUG        := FALSE;                                                 01694400010020003
  SDMASSIGNED := FALSE;                                                  01694600010020004
  CTLASSIGNED := FALSE;                                                  01694800010020004
                                                                         01695000
  WHILE FUNCTION = 0 DO                                                  01696000
  BEGIN                                                                  01697000
    START                                                                01698000
    "Enter function [Ctlload,Devload,Verifyfile,Quit]";                  0169900001.005.000
    PROMPT;                                                              01700000
    IF LL > 0 THEN                                                       01701000
    BEGIN                                                                01702000
      IF PL = "Q" THEN GO GRANDXIT;                                      01703000
      IF PL = "DB" THEN          % Debug Mode - extended displays        01704000010020003
      BEGIN                                                              01705000
        DEBUG := TRUE;                                                   01706000
        PL := * + 2;                                                     01707000
        LL := * - 2;                                                     01708000
```

D 33

```
      END                                                                    01709000010020003
      ELSE IF PL = "XM" THEN     % Express Mode - Limit displays             01710000010020003
        BEGIN                                                                01710100010020003
        EXPRESSMODE := TRUE;                                                 01710200010020003
        PL := * + 2;                                                         01710300010020003
        LL := * - 2;                                                         01710400010020003
        END;                                                                 01710500010020003
      IF PL = "C" THEN FUNCTION := LOADCTLV      ELSE                      01711000
      IF PL = "D" THEN FUNCTION := LOADEVV       ELSE                      0171200001.005.000
      IF PL = "V" THEN FUNCTION := VERIFYV       ELSE                      01713000
      SHOW("Unrecognized function selector",TRUE);                         01714000
    END; % LL > 0                                                          01715000
  END; % WHILE                                                             01716000
                                                                           01717000
                                                                           01718000
END; % SELECTOPTION                                                        01719000
$$ PAGE                                                                    01721000
                                                                           01722000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01723000
%                Epilog Procedure                                          01724000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01725000
EPILOG PROCEDURE CLEANUP;                                                  01726000
%------------------------------------------------------------------------01727000
% Purpose:                                                                 01728000
%  Clean up any hanging threads.                                           01729000
%                                                                          01730000
% Parameters:                                                              01731000
%  Epilog procedures cannot have parameters.                               01732000
%                                                                          01733000
% Globals:                                                                 01734000
%  CTLATTACHED  True:  We have to un-attach it.                            01735000
%------------------------------------------------------------------------01736000
BEGIN                                                                      01737000
  REAL RSLT;                                                               01738000
                                                                           01739000
  IF CTLASSIGNED THEN                                                      01740000
  BEGIN                                                                    01741000
    SHOW("Epilog procedure invoked to detach from CTL " C                  01742000
         CTLUNIT FOR * DIGITS,TRUE);                                       01743000
    RSLT := USERMAINTREQUEST(CTLUNIT,UNASSIGNCTLV,0,0,0,                   01744000
                             SHORTBUF,MRD,HDPRESULT);                      01745000
    IF RSLT > 0 THEN                                                       01746000
    BEGIN                                                                  01747000
      SHOW("Unable to detach from CTL " C                                  01748000
           CTLUNIT FOR * DIGITS         C                                  01749000
           ".  Error "                  C                                  01750000
           RSLT FOR * DIGITS,TRUE);                                        01751000
    END;                                                                   01752000
  END; % Control attached                                                  01753000
                                                                           01754000
  IF SDMASSIGNED THEN                                                      01755000
  BEGIN                                                                    01756000
    SHOW("Epilog procedure invoked to detach from drive " C                01757000
         CTLUNIT FOR * DIGITS,TRUE);                                       01758000
    RSLT := USERMAINTREQUEST(CTLUNIT,UNASSIGNUNITV,0,0,0,                  01759000
                             SHORTBUF,MRD,HDPRESULT);                      01760000
    IF RSLT > 0 THEN                                                       01761000
    BEGIN                                                                  01762000
      SHOW("Unable to detach from drive " C                                01763000
           CTLUNIT FOR * DIGITS          C                                 01764000
           ".  Error "                   C                                 01765000
           RSLT FOR * DIGITS,TRUE);                                        01766000
    END;                                                                   01767000
  END; % Disk attached                                                     01768000
END; % Epilog procedure                                                    01769000
$$ PAGE                                                                    01770000
                                                                           01771000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01772000
%                Outer Block                                               01773000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%01774000
                                                                           01775000
```

```
   INITIALIZE;                                                                  01776000
                                                                                01777000
MAINLOOP:                                                                       01778000
  WHILE TRUE DO                                                                 01779000
    BEGIN                                                                       01780000
    SELECTOPTION;                                                               01781000
                                                                                01782000
    CASE FUNCTION OF                                                            01783000
      BEGIN                                                                     01784000
    LOADCTLV:                                                                   01785000
        OPTODO := UNASSIGNCTLV;        % CTL   PARAMETER FOR UMR                 01785500010020004
        VERIFYFILE;                                                             01786000
        LOADCTLFW;                                                              01787000
    LOADEVV:                                                                    0178800001.005.000
        OPTODO := UNASSIGNUNITV;       % DRIVE PARAMETER FOR UMR                 01788500010020004
        VERIFYFILE;                                                             01789000
        LOADEVFW;                                                               0179000001.005.001
    VERIFYV:                                                                    01791000
        VERIFYONLY := TRUE;                                                     01791500010020004
        VERIFYFILE;                                                             01792000
        VERIFYONLY := FALSE;                                                    01792500010020004
      END;  % CASE FUNCTION OF                                                  01793000
                                                                                01794000010020004
    END;  % WHILE TRUE                                                         01795000
                                                                                01796000
GRANDXIT:                                                                       01797000
  SHOW ("<< DFAST Program is ENDING >>", TRUE);                                 01798000010010001
  IF CODE.OPEN THEN                                                             01799000
    CLOSE(CODE);                                                                01800000
  IF NOT ODTMODE THEN                                                           01801000
    CLOSE(RMT);                                                                 01802000
  CLOSE(LINE);                                                                  01803000
                                                                                01804000
END.                                                                            01805000
```